KLASSIFIZIERUNG VON PERSONENDATEN MITTELS MACHINE LEARNING:

ENTWICKLUNG EINES NAMED ENTITY RECOGNITION MODELLS ZUR UNTERSTÜTZUNG DER KONFORMITÄT VON UNTERNEHMEN MIT DEM DSG UND DER DSGVO

Bachelor Thesis

HWZ Hochschule für Wirtschaft Zürich

eingereicht bei:

Fabian Schöni

Vorgelegt von:	Lilian Bühler
Matrikelnummer:	21-521-141
Studiengang:	Bachelor of Science in Business Information Technology
Ort, Datum:	Zürich, 18.05.2025

Management Summary

Mit der fortschreitenden Digitalisierung und der Revision des Datenschutzgesetzes stehen Unternehmen vor der Herausforderung, die Einhaltung datenschutzrechtlicher Vorgaben gezielt umzusetzen. Die europäische Datenschutzgrundverordnung (DSGVO) sowie das revidierte Schweizer Datenschutzgesetz (DSG) formulieren klare Anforderungen an die Verarbeitung personenbezogener und besonders schützenswerter Daten. Die automatisierte Erkennung sensibler Informationen in unstrukturierten Texten erweist sich dabei als anspruchsvoll. Daraus ergibt sich die zentrale Frage, inwieweit technische Ansätze wie Named Entity Recognition Unternehmen bei der Einhaltung dieser Vorgaben unterstützen können.

Ziel dieser Arbeit war es, ein bestehendes NER-Modell so zu erweitern, dass neben klassischen Personendaten auch besonders schützenswerte Daten wie Religion, ethnische Herkunft und sexuelle Orientierung erkannt werden können. Hierzu wurde ein bestehendes NER-Modell mithilfe eines öffentlich verfügbaren Datensatzes sowie einem eigens erstellten synthetischen Datensatz feinjustiert.

Die Ergebnisse zeigen, dass das erweiterte Modell eine hohe Klassifikationsgenauigkeit erreicht und die neuen sensiblen Kategorien zuverlässig erkennen kann. Dies gilt insbesondere für strukturierte Texte. Mögliche Anwendungsbereiche sind die automatisierte Schwärzung sensibler Inhalte, die Klassifikation von Dokumenten und die Erweiterung bestehender Data Loss Prevention Systeme.

Gleichzeitig wurden auch Grenzen deutlich. Die Modellleistung nimmt in bei realen und unstrukturierten Texten spürbar ab, was auf die formale Struktur der verwendeten Trainingsdaten zurückzuführen ist. Für einen produktiven Einsatz ist daher eine weiterführende Evaluation mit realen Daten empfohlen.

Insgesamt zeigt die Arbeit, dass der gezielte Einsatz von NER-Modellen Unternehmen wirksam dabei unterstützen kann, datenschutzrechtliche Anforderungen effizienter umzusetzen. Die entwickelte Lösung bietet eine solide Ausgangsbasis für den weiteren Ausbau technischer Datenschutzunterstützung. Das trainierte Modell ist auf der Plattform Hugging Face unter dem Namen HuggingLil/pii-sensitive-ner-german veröffentlicht.

Inhaltsverzeichnis

1. Einleitung

1.1. Ausgangslage

Die Datenschutzgesetze in der EU und in der Schweiz wurden in den letzten Jahren grundlegend überarbeitet (EDPS, 2018). Auslöser war insbesondere der technologische Wandel, der neue Anforderungen an den Umgang mit personenbezogenen Daten mit sich bringt. Die bisherigen Regelungen boten Unternehmen nur begrenzte Orientierung im digitalen Kontext. Eine Überarbeitung war daher notwendig, um den aktuellen technologischen Entwicklungen Rechnung zu tragen. Vor diesem Hintergrund wurde im Jahr 2016 die Datenschutzgrundverordnung (DSGVO) der Europäischen Union verabschiedet.

Mit dem Aufkommen des Internets und der fortschreitenden Digitalisierung werden zunehmend mehr Daten generiert, verarbeitet und gespeichert. Für Ende 2025 wird ein Datenvolumen von 181 Zettabyte vorhergesagt (Eagar, 2023). Darin ist auch eine erhebliche Menge an personenbezogenen Daten enthalten. Früher wurden Daten oft manuell verwaltet, in Aktenordnern und Archiven. Heute existieren viele Daten ausschliesslich in digitaler Form und werden entsprechend nur digital verarbeitet. Viele Unternehmensprozesse haben sich durch die Digitalisierung verändert. Die Online-Vermarktung von Produkten beispielsweise, hat einen völlig neuen Vertriebskanal ermöglicht. Dies hat eine veränderte Arbeitsweise zur Folge und erfordert eine technische Verarbeitung der gesammelten Daten.

Die technologischen Aspekte der Datenverarbeitung wurden in den bisherigen Datenschutzgesetzen jedoch unzureichend berücksichtigt. Der Fall Cambridge Analytica verdeutlicht die Schwächen der damaligen Datenschutzgesetzgebung (Denham, 2018). Zwischen 2014 und 2016 nutzte das Unternehmen personenbezogene Daten von bis zu 87 Millionen Facebook-Nutzenden. Diese wurden ohne deren Wissen oder Einwilligung erhoben und im Rahmen eines psychografischen Profilings verwendet, um Wählende gezielt zu beeinflussen.

Ein konkreter Anwendungsfall war die Brexit-Kampagne, bei dem Cambridge Analytica personenbezogene Daten nutzte, um politische Botschaften individuell auf Persönlichkeitsprofile zuzuschneiden. Der Skandal zeigte, dass die damaligen rechtlichen Rahmenbedingungen nicht ausreichten, um algorithmisch getriebene Eingriffe in demokratischen Prozessen wirksam zu regulieren. Die DSGVO enthält nun neue Regelungen zum Profiling. Diese Anpassung steht exemplarisch für eine umfassendere Modernisierung des Datenschutzrechts.

Die neuen Datenschutzgesetze sollen die Datenhoheit den Personen zurückgeben und mehr Transparenz schaffen. Gemäss DSGVO (Datenschutzgrundverordnung, Art. 1) ist es das Ziel, die Privatsphäre zu sichern und die Sicherheit von personenbezogenen Daten zu gewährleisten. Der Datenschutz soll vor unberechtigtem Zugriff, sowohl durch den Staat als auch durch private Akteure, schützen (Berwanger, o. J.). Durch die rechtlichen Vorgaben wird ein Rahmen erstellt, um klare Regeln im Umgang mit personenbezogenen Daten zu definieren.

Ab Mai 2018 findet die Datenschutzgrundverordnung ihre Anwendung (EDPS, 2018). Mit dem Inkrafttreten der DSGVO gilt diese direkt für alle EU-Mitgliedstaaten. Die Schweiz hat auf die Datenschutzgrundverordnung der EU reagiert und verabschiedete im September 2020 ein überarbeitetes Datenschutzgesetz (DSG), welches im September 2023 in Kraft trat. Mit dem neuen Datenschutzgesetz strebt die Schweiz eine weitgehende Übereinstimmung mit internationalen Datenschutzvorgaben an, insbesondere mit der DSGVO (KMU Admin, o. J.-a).

Alle Schweizer Unternehmen inklusive kleinere und mittlere Unternehmen müssen sich an das DSG halten (KMU Admin, o. J.-b). Sofern ein Unternehmen personenbezogene Daten von Personen verarbeitet, die sich in der EU befinden, gelten zusätzlich die Vorschriften der Datenschutzgrundverordnung. Laut DSGVO (Art. 3 Abs. 2 Buchst. a und b) wird die Verarbeitung von personenbezogenen Daten dann erfüllt, wenn die Verarbeitung dazu dient, den Personen Waren oder Dienstleistungen anzubieten oder das Verhalten betroffener Personen zu beobachten.

Letzteres lässt sich wie folgt erläutern:
„Die Absicht, durch die Datenverarbeitung das Verhalten betroffener Personen in der EU zu beobachten, wird beispielsweise daran festgemacht, ob Internetaktivitäten dieser betroffenen Personen nachvollzogen (z.B. Google Analytics) und/oder Techniken zur Profilerstellung natürlicher Personen eingesetzt werden, welche beispielsweise die persönlichen Vorlieben, Verhaltensweisen oder Gepflogenheiten der Personen analysiert oder vorhergesagt werden“ (Kellerhals Carrard, 2017, S.1).

Vor diesem Hintergrund ist die DSGVO für viele Schweizer Unternehmen relevant, da bereits das Anbieten von Leistungen oder das Erfassen von Nutzerverhalten in der EU ausreicht, um in den Anwendungsbereich der Verordnung zu fallen.

1.2. Problemstellung

Um sich an die neuen Datenschutzbestimmungen halten zu können, müssen sich Unternehmen mit den rechtlichen Grundlagen und ihren bestehenden Prozessen auseinandersetzen. Hält sich ein Unternehmen nicht an die Vorgaben, kann dies zu hohen Gelstrafen führen. Gravierende Verstösse werden gemäss DSGVO (Art. 83 Abs. 5) mit einer Busse von bis zu 20 Millionen Euro oder 4% des gesamten weltweit erzielten Jahresumsatzes bestraft, es gilt jeweils der höhere Betrag. Bei weniger gewichteten Verstössen wird der Bussgeldrahmen auf bis zu 10 Millionen Euro oder auf 2% des weltweit erzielten Jahresumsatzes gesetzt, wie in der DSGVO (Art. 83 Abs. 4) definiert. Bekannte Unternehmen wie Meta, Amazon, Instagram und Google erhielten bereits Geldstrafen wegen Verstössen gegen das Gesetz (Holzhofer, 2024).

Neben möglichen Geldstrafen ist auch das Reputationsrisiko für Unternehmen relevant. Datenschutzverletzungen können das Vertrauen der Kundschaft langfristig beeinträchtigen und sich negativ auf die Wahrnehmung der Marke auswirken. Dieses Vertrauen wiederherzustellen ist oft mit hohem Aufwand verbunden.

Unternehmen sollten daher nicht nur aus rechtlichen, sondern auch aus wirtschaftlichen Gründen ein hohes Interesse daran haben, die datenschutzrechtlichen Vorgaben einzuhalten. Im Zentrum dieser Vorgaben steht der Schutz

personenbezogener Daten. Die DSGVO und das DSG unterscheiden dabei zwischen allgemeinen Personendaten und besonders schützenswerten Personendaten. Nachfolgend eine Auflistung aller Personendaten nach DSG und DSGVO:

Tabelle 1: Rechtliche Einordnung Personendaten

Personendaten	
Datenschutzgrundverordnung EU DSGVO (Art. 4 Ziff. 1)	**Datenschutzgesetz Schweiz DSG (Datenschutzgesetz, Art. 5 Buchst. a)**
„personenbezogene Daten“ alle Informationen, die sich auf eine identifizierte oder identifizierbare natürliche Person (im Folgenden „betroffene Person“) beziehen; als identifizierbar wird eine natürliche Person angesehen, die direkt oder indirekt, insbesondere mittels Zuordnung zu einer Kennung wie einem Namen, zu einer Kennnummer, zu Standortdaten, zu einer Online-Kennung oder zu einem oder mehreren besonderen Merkmalen, die Ausdruck der physischen, physiologischen, genetischen, psychischen, wirtschaftlichen, kulturellen oder sozialen Identität dieser natürlichen Person sind, identifiziert werden kann	alle Angaben, die sich auf eine bestimmte oder bestimmbare natürliche Person beziehen

Tabelle 2: Rechtliche Grundlagen besonders schützenswerte Personendaten

Besonders schützenswerte Personendaten	
Datenschutzgrundverordnung EU DSGVO (Art. 9 Abs. 1)	**Datenschutzgesetz Schweiz DSG (Art. 5 Buchst. c)**
rassische und ethnische Herkunft	Daten über die Zugehörigkeit zu einer Rasse oder Ethnie
politische Meinungen	politische Ansichten oder Tätigkeiten
religiöse oder weltanschauliche Überzeugungen	religiöse, weltanschauliche Ansichten oder Tätigkeiten

Gewerkschaftszugehörigkeit	gewerkschaftliche Ansichten oder Tätigkeiten
genetische Daten	genetische Daten
biometrische Daten	biometrische Daten, die eine natürliche Person eindeutig identifizieren
Gesundheitsdaten	Daten über die Gesundheit
Sexualleben sowie sexuelle Orientierung	Daten über die Gesundheit oder die Intimsphäre
	Daten über verwaltungs- und strafrechtliche Verfolgungen oder Sanktionen
	Daten über Massnahmen der sozialen Hilfe

Basierend auf der entsprechenden Klassifizierung gibt es unterschiedliche Vorschriften. Insbesondere in der DSGVO (Art. 9 Abs. 1) ist die Verarbeitung von besonders schützenswerten Daten untersagt. Laut DSGVO (Art. 4 Ziff. 2) ist der Begriff Verarbeitung wie folgt definiert: Jedem mit oder ohne Hilfe automatisierter Verfahren ausgeführten Vorgang oder jede solche Vorgangsreihe im Zusammenhang mit personenbezogenen Daten wie das Erheben, das Erfassen, die Organisation, das Ordnen, die Speicherung, die Anpassung oder Veränderung, das Auslesen, das Abfragen, die Verwendung, die Offenlegung durch Übermittlung, Verbreitung oder eine andere Form der Bereitstellung, den Abgleich oder die Verknüpfung, die Einschränkung, das Löschen oder die Vernichtung.

Das bedeutet für Unternehmen, dass sie keinerlei Daten erfassen oder speichern dürfen, in denen besonders schützenswerte Daten enthalten sind. Ein Beispiel wäre, dass ein Kunde (Auslandschweizer mit dem Wohnsitz in Österreich) seine Steuererklärung, auf der die Religion aufgeführt wird, seiner Schweizer Bank schickt. Diese soll anschliessend eine Finanzierungsprüfung vornehmen. Im Prozess der Bank ist definiert, dass die Unterlagen der Kunden erst einmal im elektronischen Kundendossier abgelegt werden. Gemäss DSGVO zählt die Speicherung des Dokuments bereits zur Verarbeitung, was in diesem Fall untersagt ist. Unternehmen sollten somit vorgängig überprüfen, welche Daten sie in ihrem Unternehmen verarbeiten wollen und dürfen. Es könnte beispielsweise eine Eingangsprüfung vorgenommen werden, bevor Daten im Unternehmen verarbeitet werden.

Besonders schützenswerte Personendaten dürfen nicht gespeichert werden. Daher sollten sich alle Unternehmen, die der DSGVO unterliegen, einen Überblick über ihre bestehenden Datenbestände verschaffen und diese gegebenenfalls bereinigen. Im DSG wird die Verarbeitung besonders schützenswerter Daten nicht untersagt, jedoch wird laut DSG (Art. 6 Abs. 7) die ausdrückliche Einwilligung verlangt. Ein weiterer wichtiger Aspekt aus der Datenschutzgrundverordnung ist die zweckgebundene Datenbeschaffung. Laut DSG (Art. 6 Abs. 3) dürfen nur Personendaten beschafft werden, welche für einen bestimmten Zweck vorgesehen sind. Die Verarbeitung der Personendaten muss zwingend mit dem Zweck übereinstimmen. Die Zweckgebundenheit wird ebenfalls von der DSGVO gefordert und ist in der DSGVO (Art. 6 Ziff. 1 Abs. a und b) nur dann zweckmässig, wenn die betroffene Person einwilligt oder die Verarbeitung für die Erfüllung des Vertrags notwendig ist. Unternehmen müssen genau definieren, für welche Geschäfte welche Informationen notwendig sind, um sich an die Gesetze zu halten.

Es sind somit zwei Hauptkomponenten notwendig, um die datenschutzrechtliche Konformität sicherzustellen. Zum einen sollten sich Unternehmen einen Überblick darüber verschaffen, welche personenbezogenen Daten bereits vorhanden sind, um deren Handhabung korrekt zu gestalten. Zum anderen sollten technische und organisatorische Massnahmen sicherstellen, dass keine unzulässigen oder unerwünschten Daten in die Systeme gelangen. Beide Anforderungen sind mit erheblichem Aufwand verbunden, insbesondere bei grossen und unstrukturierten Datenbeständen. Eine manuelle Sichtung aller Daten innerhalb kurzer Zeit ist in der Praxis kaum realisierbar.

Mit dem Aufkommen von Machine-Learning-Verfahren eröffnen sich jedoch neue Möglichkeiten, diese Herausforderung automatisiert zu bewältigen. Insbesondere Named Entity Recognition (NER) kann dabei unterstützen, personenbezogene und besonders schützenswerte Daten in Texten zu identifizieren und zu klassifizieren.

Daraus ergibt sich die nachfolgende Forschungsfrage.

1.3. Forschungsfrage

Wie kann Named Entity Recognition zur Klassifizierung personenbezogener Daten eingesetzt werden, um Unternehmen bei der DSGVO- und DSG-Konformität zu unterstützen?

1.4. Zielsetzung und Abgrenzung

Das Ziel dieser Arbeit ist es, ein Machine-Learning-Modell zu entwickeln und zu trainieren, das personenbezogene Daten aus einem Textabschnitt erkennen und entsprechend kategorisieren kann. Ein Beispiel wäre der Satz: „Max hat seine Konfirmation am 14. April 2024." In diesem Fall deutet „Konfirmation" auf die Religion hin, die dann als „Religion" mit dem Inhalt „Konfirmation" vom Modell erkannt und gekennzeichnet wird. Ebenfalls sollte der Vorname „Max" vom Modell erkannt und entsprechend klassifiziert werden. Zur Erreichung dieses Ziels wird die Methode der Named Entity Recognition (NER) angewendet. Vor der technischen Umsetzung werden relevante Methoden und Vorgehensweisen theoretisch aufgearbeitet und dokumentiert.

Die erste Version des Modells wird nicht alle Informationen umfassend erkennen können. Stattdessen wird der Fokus auf ausgewählte Datenkategorien gemäss DSG und DSGVO gesetzt, die vorab festgelegt werden. Die beiden Gesetze bieten die Grundlage für diese Arbeit, weitere beispielsweise Regionale Verordnungen und Gesetze finden hier keine Anwendung. Das Modell soll für verschiedene Anwendungszwecke nutzbar sein. Die konkrete Implementierung in einen Real-World-Case ist nicht Bestandteil dieser Arbeit. Allerdings werden mögliche Anwendungsbeispiele dokumentiert.

Ziel ist es, dass das Modell primär als Klassifikationswerkzeug fungiert, nicht als Data Loss Prevention-Massnahme. Dennoch könnte es Unternehmen zusätzlich dabei unterstützen, ungewollten Datenabfluss zu verhindern. Das Modell ist primär für den Einsatz in deutschsprachigen Unternehmen konzipiert, weshalb für das Training deutschsprachige Datensätze verwendet werden. In dieser Arbeit werden keine vollständigen Dokumente mit unterschiedlichen Dateiformaten verarbeitet, stattdessen basiert das Training auf einzelnen Textfragmenten. Um datenschutzrechtliche Risiken

auszuschliessen und das Modell frei nutzbar zu machen, wird es auf synthetischen Daten trainiert.

Das Modell bietet gegenüber bestehenden Softwarelösungen die Möglichkeit, es noch weiter zu trainieren und anzupassen. Die Nutzung des Modells kann frei gewählt werden. Darüber hinaus lässt sich die Genauigkeit des Modells erhöhen, wenn es mit unternehmensinternen Daten weitertrainiert wird. Im Gegensatz zu bestehenden Lösungen steht das Modell öffentlich zur Verfügung und verursacht keine Lizenzkosten.

2. Theoretische Grundlagen

Im Folgenden werden die theoretischen Grundlagen erläutert, um ein gemeinsames Verständnis zu etablieren. Es werden die technischen Aspekte behandelt, welche als Basis für die anschliessende Umsetzung, dem Fine Tuning eines Machine Learning Modells dienen.

2.1. Machine Learning

Machine Learning (ML) ist ein Teilgebiet der Artificial Intelligence (AI) und befasst sich mit der automatisierten Erkennung von Mustern in Daten (Thareja, 2024). Auf dieser Basis können Modelle eigenständig Vorhersagen treffen. Während des Trainings trifft das Modell bereits Entscheidungen, die mit den tatsächlichen Ergebnissen verglichen werden. Dadurch lernt es, welche Vorhersagen korrekt gewesen wären, und optimiert seine Mustererkennung fortlaufend. Beim klassischen Machine Learning werden die Modelle durch manuelles Feature Engineering unterstützt. Dabei gibt der Mensch vor, welche Merkmale für die Mustererkennung relevant sind.

Für das Training von ML-Modellen werden Daten in drei Kategorien unterteilt: Trainingsdaten, Validierungsdaten und Testdaten (Taulli, 2025). Die Trainingsdaten dienen dazu, Muster zu erkennen. Die Validierungsdaten kommen während des Trainings zum Einsatz, um die Modellleistung zu überprüfen und zu verbessern. Die Testdaten werden erst nach dem Training verwendet, um die finale Modellleistung anhand von Metriken wie Accuracy, Precision, Recall oder F1-Score zu bewerten. Auf die verschiedenen Evaluationsmetriken wird im Kapitel Performance Evaluation genauer eingegangen.

Die erlernten Muster werden im Modell gespeichert, wodurch es auch nach dem Training auf das erworbene Wissen zugreifen kann (Thareja, 2024). Ein trainiertes Modell kann durch sogenanntes Fine Tuning weiter verbessert werden, ohne den Lernprozess vollständig neu zu starten (Amaratunga, 2023).

Machine Learning wird in drei Hauptkategorien unterteilt: Supervised Learning, Unsupervised Learning und Reinforcement Learning (Thareja, 2024). Im Rahmen dieser Arbeit wird ausschliesslich Supervised Learning verwendet.

2.2. Supervised Learning

Das Training des Algorithmus wird beim Supervised Learning mittels gelabelten Trainingsdatensätzen vorgenommen (LLC, 2025a). Diese bestehen aus Eingabedaten und den dazugehörigen Zielwerten. Somit wird der Input bereits mit dem zu erzielenden Output verknüpft. Aufgrund dieser Verbindung zwischen Eingabewert und Zielkategorie kann das Modell Zusammenhänge erkennen und verinnerlichen. Durch das Lernen der Verbindungen wird das Modell befähigt, das Gelernte auch für neue Daten anzuwenden.

Eine Art des Supervised Learning ist die Klassifikation des Inputs (Barua et al., 2024). Dabei wird eine Vorhersage gemacht, zu welcher Klasse der eingegebene Input gehören könnte. Das Modell erlernt die Zuordnung zu den vordefinierten Labels im Rahmen des Trainingsprozesses. Dadurch ist das Modell in der Lage, die gelernten Labels auch auf neue, zuvor unbekannte Eingabedaten anzuwenden. Beispiele für Klassifikations-Modelle sind die Sentimentanalyse, die Klassifikation von E-Mails (Spam / Kein Spam) oder die Einordnung von Bildern (Hund oder Katze).

Supervised Learning Methoden ermöglichen es, ein Modell auf spezifische Aufgaben zu trainieren (Lammle & Robb, 2024). Dadurch kann für die entsprechende Aufgabe eine bessere Leistung erzielt werden. Ein Nachteil des Verfahrens ist es jedoch, dass gelabelte Trainingsdaten notwendig sind. Ein weiterer Nachteil besteht darin, dass das Modell gegenüber unbekannten Daten eine schlechtere Performance aufweist.

2.3. Deep Learning

Deep Learning ist im Gegensatz zum Machine Learning nicht auf manuelles Feature Engineering angewiesen (Tabrizi, 2025). Dem Modell wird also nicht mehr gesagt, worauf es sich fokussieren soll, sondern es entdeckt und optimiert die relevanten Merkmale während des Trainingsprozesses selbstständig.

Die gesamte Modellarchitektur wird als neuronales Netz bezeichnet, das aus mehreren Verarbeitungsschichten besteht. Die einzelnen Verarbeitungseinheiten innerhalb dieser Schichten werden als Neuronen bezeichnet. Jede Schicht verarbeitet dabei die

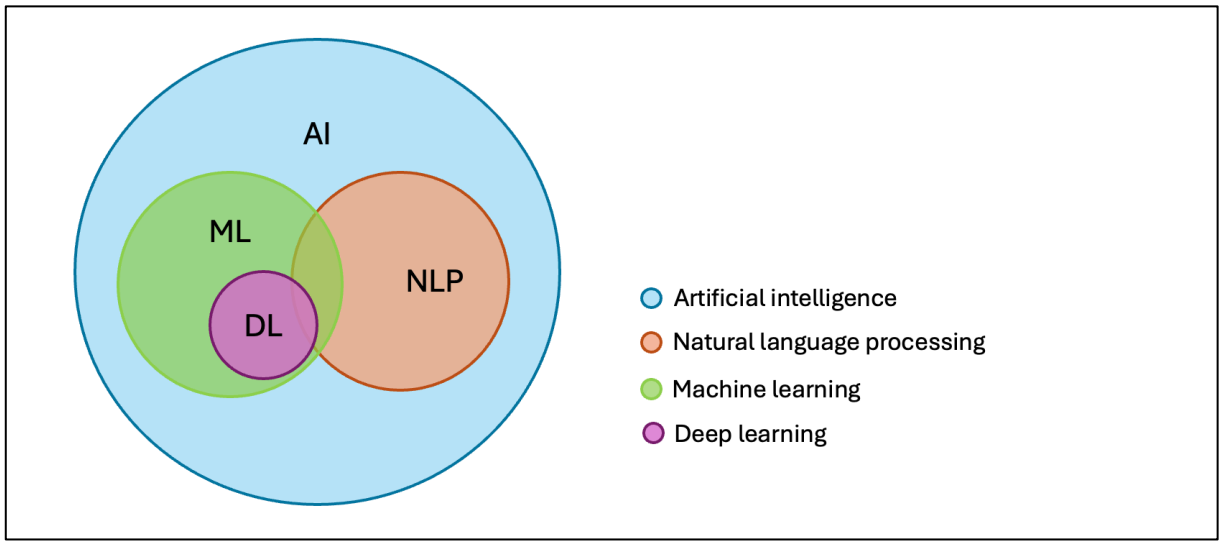
Ausgaben der vorherigen Schicht. Sobald es sich um ein vielschichtiges Modell handelt, wird es auch als Deep Neural Network bezeichnet.

Deep Learning Modelle eignen sich aufgrund ihrer Architektur besonders gut für komplexe Aufgaben, wie der Bild- und Spracherkennung oder der Verarbeitung natürlicher Sprache (Natural Language Processing) (Tabrizi, 2025). Deep Learning Modelle werden mit zunehmender Datenmenge und zunehmender Rechenleistung immer performanter und können sich stetig weiterentwickeln, wohingegen Machine-Learning-Modelle irgendwann ihr Leistungsplateau erreichen. Der Grund dafür ist, dass Deep Learning Modelle mit mehr Daten und höherer Rechenleistung in der Lage sind, komplexere Muster zu erkennen und somit auch genauere Vorhersagen treffen können.

2.4. Natural Language Processing

Natural Language Processing (NLP) gilt, wie auch das Machine Learning, als Unterkategorie der Artificial Intelligence (AI) (Amaratunga, 2023).

Abbildung 1: Zusammenspiel zwischen AI, ML, DL und NLP



Quelle: In Anlehnung an Thareja (2024)

NLP ermöglicht es Computern, die menschliche Sprache zu verstehen und zu interpretieren (Amaratunga, 2023). Die Konzepte und Algorithmen der NLP-Forschung dienen als Grundlage für Large Language Models (LLM). Large Language Models

wiederum verwenden Deep Learning Methoden, um sprachliche Zusammenhänge in grossen Textmengen zu erkennen und auf komplexe Aufgaben anzuwenden.

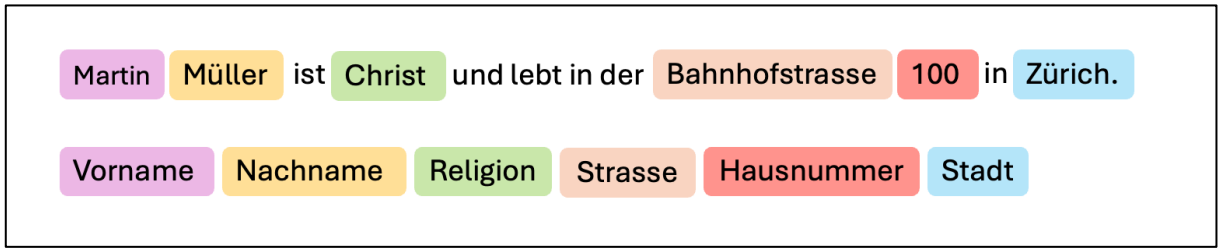
Large Language Models (LLM) basieren auf der Transformer-Architektur, einer speziellen Form von Deep Learning Modellen, die für die Verarbeitung von Sprachdaten entwickelt wurde (Khraisha, 2024). LLMs werden auf der Basis von mehreren Milliarden Wörtern trainiert und können dadurch in zahlreichen Anwendungsbereichen eingesetzt werden. Bekannte Beispiele für LLMs sind OpenAIs GPT-Reihe, Googles BERT (Bidirectional Encoder Representations from Transformers) sowie Metas LLaMA und Claude (Khraisha, 2024).

Typische Aufgaben für NLP sind: Textgenerierung, Übersetzungen, das Erstellen von Zusammenfassungen, die Beantwortung von Fragen, Sentimentanalysen sowie die Klassifikation von Texten (Erik, 2025).

2.5. Named Entity Recognition

Named Entity Recognition (NER) ist eine weitere zentrale Aufgabe des Natural Language Processing (NLP) (LLC, 2025b). Dabei handelt es sich um die Identifikation von Entitäten innerhalb eines Textes. Diese Textbestandteile (Tokens) werden vordefinierten Klassen zugeordnet. Es ermöglicht somit eine gezielte Informationsgewinnung als auch die strukturierte Analyse verschiedener Textdaten.

Abbildung 2: Angewendete Named Entity Recognition



Die Abbildung zeigt ein Beispiel für die Anwendung von Named Entity Recognition im Zusammenhang mit der Erkennung von Personendaten. Im dargestellten Satz wurden verschiedene Entitäten erkannt und ihren entsprechenden Labels zugeordnet. Die Labels wie Vorname, Nachname, Religion, Strasse, Hausnummer und Stadt wurden

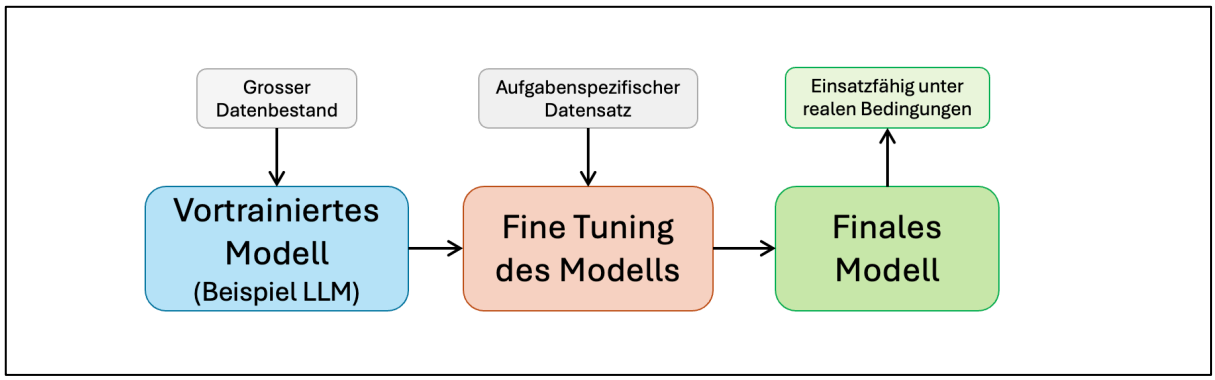
dem NER-Modell vorgängig beigebracht. Das Modell kann somit auf beliebigen Text angewendet werden.

Named Entity Recognition kann auf unterschiedliche Weise umgesetzt werden (Khraisha, 2024). Bei lexikonbasierten Verfahren werden Entitäten anhand vordefinierter Listen oder Wörterbücher erkannt. Regelbasierte Verfahren nutzen manuell definierte Regeln und Muster, häufig in Form von regulären Ausdrücken (Regex), um bestimmte Begriffe oder Strukturen im Text zu identifizieren. Neben diesen klassischen Ansätzen werden zunehmend auch Deep Learning Methoden eingesetzt, wobei insbesondere der Einsatz von Large Language Models (LLMs) an Bedeutung gewinnt.

2.6. Fine Tuning eines Modells

Um ein bestehendes Modell auf eine spezifische Domäne anzupassen, kann es feinjustiert werden (Khraisha, 2024). Beim Fine Tuning wird das bestehende Modell auf einem spezifischen Datensatz weitertrainiert, sodass es sein vorhandenes Wissen nutzt und sich gezielt an die Fachsprache, die Syntax und den Kontext der Zieldomäne anpasst.

Abbildung 3: Fine Tuning eines vortrainierten Modells



Die Abbildung zeigt den typischen Ablauf bei der Entwicklung eines einsatzfähigen Modells mittels Fine Tuning. Ausgangspunkt ist ein vortrainiertes Modell, welches auf einem grossen allgemeinen Datensatz trainiert wurde, dies kann zum Beispiel ein Large Language Model wie Bert oder GPT sein (Alammar & Grootendorst, 2024). Dieses Modell wird im Anschluss mit einem aufgabenspezifischen Datensatz

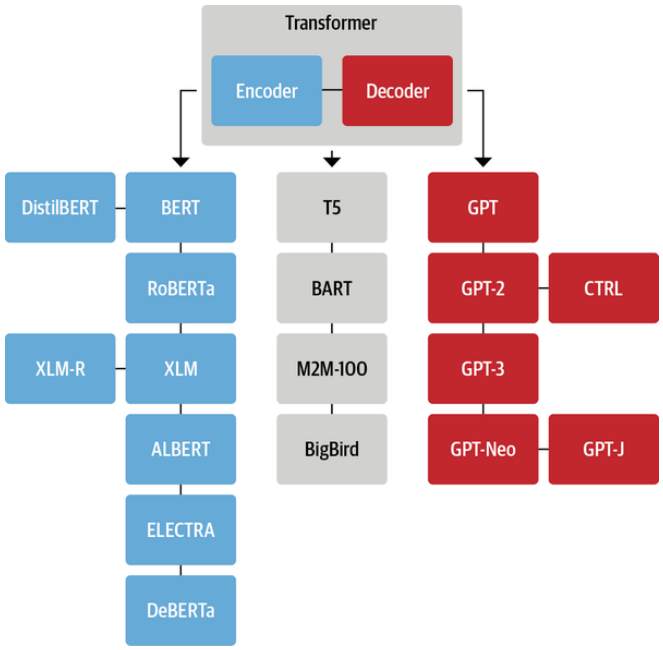
weitertrainiert. Damit wird ermöglicht, das Modell auf spezifische Anforderungen zu trainieren, wie etwa auf eine bestimmte Domäne. Durch das Fine Tuning wird ein finales Modell geschaffen, das auf die spezifische Aufgabe optimiert ist und unter realen Bedingungen bessere Ergebnisse liefert.

2.7. Transformer Modell

Transformer Modelle gehören zur Familie der künstlichen neuronalen Netze und sind ein Teilbereich des Deep Learning. Sie wurden speziell für die Verarbeitung von sequenziellen Daten entwickelt, wie beispielsweise natürliche Sprache oder Text. Transformer-Modelle arbeiten mit speziellen Schichten, den sogenannten Attention-Layern (Transformer-Modelle - HUG, o. J.). Diese Layer werden verwendet, um dem Modell zu zeigen, worauf es ein besonderes Augenmerk bei den Inputdaten haben soll.

Transformer Modelle können in verschiedene Arten unterteilt werden: Encoder, Decoder und Encoder-Decoder Modelle (Transformer-Modelle - HUG, o. J.). Die Abbildung veranschaulicht die Einteilung verschiedener Transformer Modelle anhand bekannter Beispiele.

Abbildung 4: Darstellung der bekanntesten Transformer Modelle



Quelle: Tunstall et al. (2022, o.S.)

Im Rahmen dieser Arbeit wird ein reines Encoder Model verwendet, welches sich für den Anwendungsfall von Named Entity Recognition eignet.

Encoder Modelle erstellen eine numerische Darstellung des Inputs, indem sie wichtige Merkmale (Features/Embeddings) extrahieren (Transformer-Modelle - HUG, o. J.). Features beziehungsweise Embeddings sind numerische Darstellungen von Eigenschaften oder Bedeutungen der Eingabedaten, die das Modell anschliessend für weitere Verarbeitungsschritte nutzt. Dadurch kann das Modell semantische Zusammenhänge und die strukturelle Beziehung zwischen den Eingabewörtern erfassen und als Grundlage für weitere Verarbeitungsschritte nutzen.

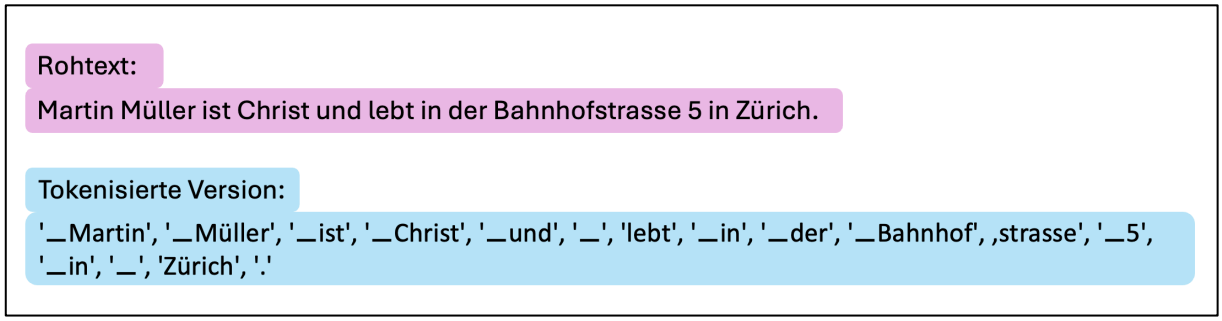
2.8. Piiranha-Modell

Im Zuge dieser Arbeit wird ein Fine Tuning des Piiranha-Modells vorgenommen (piiranha-v1 - HUG, 2025). Das Modell ist eine feinjustierte Variante des mdberta-v3-base Checkpoints und wurde explizit für die Erkennung personenbezogener Daten trainiert. Das von Piiranha verwendete vortrainierte Modell mdeberta-v3-base ist eine multilinguale Version von DeBERTa und nutzt dieselbe Struktur (mdeberta-v3-base - HUG, o. J.). DeBERTa (Decoding-enhanced BERT with disentangled attention) ist ein reines Encoder Modell, das sich besonders gut für Aufgaben wie die Named Entity Recognition eignet. DeBERTa trennt die Verarbeitung von Positions- und Inhaltsinformationen, wodurch beide Aspekte unabhängig voneinander analysiert werden (He et al., 2021). Dadurch wird die Wortbedeutung nicht durch die Position beeinflusst und umgekehrt. Andernfalls könnte die Modellleistung bei NER-Aufgaben leiden, da Entitäten fälschlicherweise aufgrund ihrer Position im Satz klassifiziert werden, anstatt aufgrund ihrer tatsächlichen Bedeutung.

2.9. Tokenisierung

Die Tokenisierung ist der erste Teil der Datenvorverarbeitung für NLP-Aufgaben (Erik, 2025). Bei der Tokenisierung werden die Sätze oder Wörter in einzelne Teile (Subwords) unterteilt. Diese Zerlegung erlaubt es, auch unbekannte oder seltene Wörter in sinnvolle Bestandteile aufzubrechen.

Abbildung 5: Tokenisierung eines Satzes mittels Piiranha-Tokenizer



Die Abbildung veranschaulicht, wie ein Satz in kleinere Texteinheiten (Tokens) zerlegt wird, um ihn für ein NLP-Modell verarbeitbar zu machen. In der dargestellten Tokenisierung werden Subwords durch Unterstriche (_) markiert, die den Beginn eines neuen Wortbestandteils anzeigen. Andere Tokenizer, wie beispielsweise bei BERT-Modellen, verwenden Doppelkreuze (##), um Fortsetzungen von Wörtern zu kennzeichnen (Tokenizer - HUG, o. J.).

Durch die Aufteilung in Subwords bleibt der Wortschatz kompakt und das Modell kann flexibel auf neue Eingaben reagieren (Erik, 2025). Durch die Tokenisierung wird der Text in eine Form gebracht, die das Modell verarbeiten kann. Anstelle von langen, komplexen Sätzen arbeitet das Modell mit Subwords, die es ermöglichen, Sprache effizient und genau zu analysieren. Da neuronale Netze nicht direkt mit rohem Text arbeiten können, sondern numerische Repräsentationen benötigen, ist die Tokenisierung ein notwendiger Schritt. Nach der Tokenisierung wird jedem Token eine eindeutige Nummer aus dem Vokabular zugewiesen, sodass das Modell die Eingabedaten in numerischer Form verarbeiten kann. Eine saubere Tokenisierung ist das A und O für NLP, da dies die Genauigkeit des Modells essenziell beeinflusst.

Welche Art von Tokenisierung verwendet wird, hängt vom verwendeten Modell ab (Crowe et al., 2024). Dabei ist es wichtig, dass die gewählte Tokenisierung mit dem ursprünglichen Tokenizer übereinstimmt, mit dem das Modell trainiert wurde. Dies ist relevant, da es je nach Tokenizer Unterschiede gibt, vor allem bei der Geschwindigkeit, dem Umgang mit Leerzeichen oder der Unterstützung verschiedener Sprachen. Transformer basierte NLP-Modelle benötigen spezielle Tokens, um den Beginn oder das Ende des Modelleingabefeldes zu kennzeichnen (Raschka, 2025). Bei der

Vorbereitung von Daten für die Batchverarbeitung kommen zusätzlich sogenannte Pad-Tokens zum Einsatz. Diese dienen dazu, die Eingabesequenzen innerhalb eines Batches auf eine einheitliche Länge zu bringen, um die parallele Verarbeitung durch das Modell zu ermöglichen.

2.10. Synthetische Daten

Synthetische Daten sind künstlich erzeugte Daten, die gezielt für einen bestimmten Anwendungszweck generiert werden (Huyen, 2025). Dadurch kann ein definierter Qualitätsstandard sichergestellt werden. Die Qualität von Machine-Learning-Modellen hängt direkt von den verwendeten Trainingsdaten ab. Vor diesem Hintergrund gewinnen synthetische Daten zunehmend an Bedeutung. Sie werden eingesetzt, um reale Daten zu simulieren, und können diese entweder ergänzen oder vollständig ersetzen. Besonders vorteilhaft ist ihr Einsatz, wenn reale Daten nicht verfügbar sind, aus Datenschutzgründen nicht genutzt werden dürfen oder die Beschaffung mit erheblichem Aufwand verbunden ist.

2.11. Performance Evaluation

Die Evaluation ist essenziell, um die Genauigkeit des Models zu bestimmen. Ein Named-Entity-Recognition-Modell sollte in der Lage sein, alle Entitäten zu erkennen und diese ihren Entitätstypen zuzuweisen (Khraisha, 2024).

Um die Performance zu errechnen sind folgende Werte notwendig, diese wird auch als Confusion Matrix bezeichnet:

Abbildung 6: Confusion Matrix

Confusion Matrix		
	tatsächliche Entität	**keine tatsächliche Entität**
Erkannt als Entität	✓ True positive	✗ False positive
Nicht als Entität erkannt	✗ False negative	✓ True negative

Quelle: In Anlehnung an Khraisha (2024)

Die Confusion Matrix weisst vier Werte aus: True Positive (TP), False Positive (FP), False Negative (FN) und True Negative (TN) (Khraisha, 2024). True Positives sind Entitäten, welche vom NER-Modell korrekt als solche erkannt wurden. Bei den False Positives handelt es sich um Wörter oder Inhalte, die vom Modell fälschlicherweise als Entitäten erkannt wurden, obwohl sie tatsächlich keine sind. False Negative Werte sind Entitäten, welche vom Modell nicht erkannt wurden. True Negatives sind keine Entitäten und wurden vom Modell auch keiner Entität zugewiesen.

Um die Werte der Confusion Matrix zu berechnen, braucht man den Ground Truth Datensatz mit den tatsächlichen, korrekten Labels (Khraisha, 2024). Dies ist der Testdatensatz, indem die vorhergesagten Werte des Modells mit den effektiven Werten aus der Ground Truth abgeglichen werden.

Aus den Werten der Confusion Matrix können dann spezifische Metriken errechnet werden, um detaillierte Aussagen zur Performance des Modells zu treffen (Khraisha, 2024). Der F1 Score wird häufig als Default Metrik verwendet, da er ein ausgewogenes Verhältnis zwischen Precision und Recall errechnet. Allerdings sollten die Metriken Accuracy, Precision und Recall ebenfalls ermittelt werden, um konkretere Aussagen bezogen auf den Anwendungsfall zu treffen. Nachfolgend werden die erwähnten Metriken genauer erklärt.

2.11.1. Accuracy

Die Accuracy beantwortet folgende Frage: „Wie viele der vorgenommenen Klassifikationen waren korrekt (True Positive und True Negative)?“ (Khraisha, 2024). Es bedeutet im Kontext von NER, wie gut das Model Entitäten und Nicht-Entitäten erkennen kann. Die Accuracy ist dann ein aussagekräftiger Wert, wenn die False Positives und False Negatives als gleich ungünstig bewertet werden.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

Jedoch bietet die Accuracy möglicherweise kein umfassendes Bild der Performance. Vor allem bei Datensätzen mit einer unausgewogenen Verteilung, kann dies zu einer Fehleinschätzung führen. Wenn der Datensatz überwiegend aus Tokens besteht, die

keine Entität darstellen, kann ein Modell eine hohe Accuracy erreichen, indem es hauptsächlich Nicht-Entitäten vorhersagt. In einem solchen Fall entsteht ein verzerrtes Bild der tatsächlichen Leistungsfähigkeit, da echte Entitäten kaum erkannt werden. Um die Modellleistung realistisch einzuschätzen, sind Precision, Recall und F1-Score daher besser geeignet.

2.11.2. Precision

Die Precision misst den Anteil der korrekten positiven Vorhersagen, an allen positiven Vorhersagen: „Wie viele Instanzen von allen die als Entität erkannt wurden sind auch tatsächlich korrekt?“ (Khraisha, 2024). Sollte der Precision Wert niedrig sein, deutet es darauf hin, dass das Modell viele False Positives generiert. Der Wert ist bedeutend, wenn fälschlicherweise erkannte Entitäten dem Anwendungsfall entgegenwirken.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Die Precision ist dann relevant, wenn es beispielsweise um Erkennung von Spam E-Mails geht, hierbei muss sichergestellt werden, dass legitime Mails nicht fälschlicherweise als Spam erkannt werden, damit das Business nicht eingeschränkt wird (LLC, 2025a).

2.11.3. Recall

Der Recall misst die Rate der korrekt erkannten tatsächlichen Entitäten (True Positive Rate) (Khraisha, 2024). Der Wert beantwortet die Frage: „Wie viele der tatsächlich positiven Fälle wurden erkannt?“. Ein niedriger Wert zeigt an, dass das Modell viele False Negatives erzeugt. Dieser Wert ist relevant, sobald es schwerwiegende Konsequenzen zufolge hat, wenn Werte nicht erkannt wurden.

$$Recall = \frac{TP}{TP + FN}$$

Der Recall spielt bei der Früherkennung von Krankheiten eine wichtige Rolle (LLC, 2025a). Hierbei ist es wichtig, dass keine Diagnosen übersehen werden, um eine frühzeitige Behandlung möglich zu machen.

2.11.4. F1 score

Der F1 Score ist das Mittel von Precision und Recall (Khraisha, 2024). Dieser wird verwendet, wenn die Klassenverteilung im Datensatz unausgewogen ist. Der Wert ist relevant, wenn sowohl False Positiv als auch False Negative erkannte Werte ins Gewicht fallen. Das trifft insbesondere auf Fälle zu, in denen ein Grossteil der Daten keine Entitäten enthält.

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

2.12. Overfitting

Overfitting findet dann statt, wenn sich das Modell zu gut auf die Trainingsdaten anpasst (Géron, 2019). Das kann passieren, da komplexe Modelle wie Deep Neural Networks feine Muster in den Daten erkennen können. Wenn der Trainingsdatensatz jedoch zu viel Grundrauschen (Noise) enthält oder dieser zu klein ist, erkennt das Modell mit hoher Wahrscheinlichkeit das Rauschen selbst. Rauschen bezeichnet zufällige Muster in den Trainingsdaten, die keinen echten Informationsgehalt haben und nicht auf neue Daten übertragbar sind. Das Resultat von Overfitting ist, dass das Modell eine schlechte Performance auf neuen Daten hat, weil es sich zu sehr an die speziellen Muster und das Rauschen der Trainingsdaten angepasst hat.

Overfitting wird meist durch zu komplexe Modelle, zu kleine oder verrauschte Datensätze oder zu langes Training verursacht (Géron, 2023). Erkennbar wird Overfitting häufig am Verlauf des Training Loss und des Validation Loss: Der Training Loss misst den Fehler zwischen den Modellvorhersagen und den tatsächlichen Werten im Trainingsdatensatz (Bahree & Boyd, 2024). Ein niedriger Wert deutet darauf hin, dass das Modell die Trainingsdaten gut abbildet. Der Validation Loss hingegen zeigt, wie gut das Modell auf neue, bislang ungesehene Daten generalisiert. Er wird nach jeder Epoche anhand der Fehler im Validierungsdatensatz berechnet. Wenn der Validation Loss steigt, obwohl der Training Loss weiter sinkt, ist dies ein typisches Anzeichen für Overfitting. Das Modell beginnt in diesem Fall, sich auf irrelevante Details der Trainingsdaten zu spezialisieren, anstatt generalisierbare Muster zu lernen.

Um Overfitting zu vermeiden, können verschiedene Methoden eingesetzt werden. Der Dropout deaktiviert während des Trainings zufällig Neuronenverbindungen, L2-Regularisierung begrenzt die Grösse der Gewichte (Géron, 2023). Early Stopping beendet das Training, sobald die Performance auf Validierungsdaten abnimmt. Eine grössere Datenmenge kann das Modell dabei unterstützen, verallgemeinerbare Muster zuverlässiger zu erfassen.

2.13. Catastrophic Forgetting

Das Catastrophic Forgetting beschreibt ein Phänomen, bei dem es um das plötzliche Vergessen des vorher gelernten geht (Brown & Zai, 2020). Beispielsweise kann es passieren, dass ein NER-Modell nach dem Fine Tuning alte Labels nicht mehr erkennt, wenn die neuen Inputdaten ausschliesslich zusätzliche Labels enthalten. In solchen Fällen konzentriert sich das Modell zu stark auf die neuen Informationen und verlernt das bisher Gelernte. Um das Vergessen zu verhindern, gibt es verschiedene Methoden. Typische Ansätze sind Replay, Distillation, Regularization oder Parameter Expansion (Pai, 2025).

Replay ist eine der einfachsten Methoden, um Catastrophic Forgetting zu reduzieren (Pai, 2025). Dabei werden Trainingsdaten des ursprünglichen Modells gespeichert und gemeinsam mit den neuen Trainingsdaten verwendet. Dadurch kann der sogenannte Data Drift abgeschwächt werden. Dieser Begriff bezeichnet Veränderungen in den Eingabedaten im Vergleich zu jenen Daten, auf denen das Modell ursprünglich trainiert wurde.

Distillation verwendet einen älteren Modell-Checkpoint und vergleicht während des Trainings die Kullback-Leibler-Divergenz (KL-Divergenz) zwischen den alten und den neuen Repräsentationen (Pai, 2025). Die KL-Divergenz ist ein Mass, welches misst wie unterschiedlich zwei Wahrscheinlichkeitsverteilungen sind. Abweichungen des neuen Modells werden bestraft.

Regularization hindert das Modell daran sich beim Lernen stark zu verändern (Pai, 2025). Wenn also grosse Anpassungen in den Gewichten, die das Verhalten des Modells steuern, vorgenommen werden, werden diese mit einer Strafkomponente im

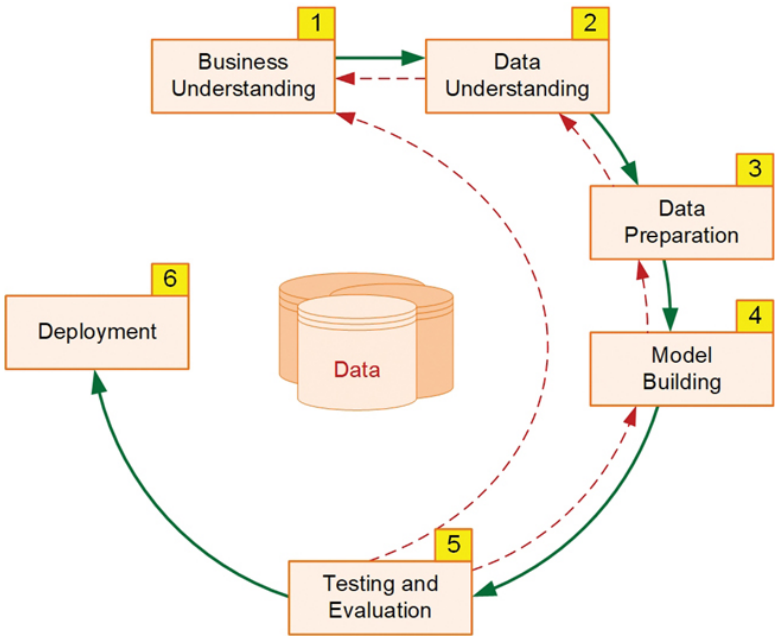
Trainingsprozess belegt. So wird das Modell dazu gezwungen, möglichst viel vom bereits gelernten Wissen beizubehalten.

Parameter Expansion fügt dem Modell während des Trainings zusätzliche Kapazitäten hinzu (Pai, 2025). Dies wird entweder mittels Erhöhung der Anzahl Neuronen pro Schicht gemacht oder mittels hinzufügen zusätzlicher Schichten. Durch diese Anpassung kann neues Wissen erlernt werden, ohne bestehendes Wissen zu überschreiben.

2.14. CRISP-DM

Das CRISP-DM Model ist ein Prozessstandard, welcher unter anderem auch für Data Science Projects verwendet wird (Delen, 2021). Der Cross-Industry Standard Process for Data Mining enthält die Phasen Business Understanding, Data Understanding, Data Preparation, Modeling, Testing & Evaluation und Deployment.

Abbildung 7: CRISP-DM Prozess



Quelle: Delen (2021, o.S.)

Obwohl die Schritte im Diagramm entlang der grünen Linie linear dargestellt werden, verläuft der Prozess in der Praxis häufig iterativ, was durch die rote gestrichelte Linie veranschaulicht wird (Delen, 2021). Es ist häufig der Fall, dass zwischen den Schritten

hin und hergewechselt wird. Da jede Phase von der vorhergehenden Abhängig ist, sollte der jeweils vorherigen Phase die nötige Aufmerksamkeit gegeben werden.

2.14.1. Business Understanding

Das Business Understanding dient als Grundlage des CRISP-DM Modells (Sarkar et al., 2018). Ein fundamentales Verständnis der Problemstellung, sowie auch der vorhandenen Datenbasis ist zwingend notwendig. Ohne ein ausreichendes Business Understanding kann das Modeling nicht effektiv umgesetzt werden. Zuerst müssen die Problemstellung und die darauffolgende Zielsetzung klar sein, um dies mittels Data Mining lösen zu können.

2.14.2. Data Understanding

Beim Data Understanding werden die vorhandenen Daten systematisch analysiert (Sarkar et al., 2018). Das Sammeln der Daten, die deskriptive Analyse und die Überprüfung der Datenqualität sind ebenfalls Teil dieser Phase. Ziel ist es frühzeitig potenzielle Verzerrungen zu erkennen und Entscheidungen zu treffen, welche für die Data Preparation Phase notwendig sind.

2.14.3. Data Preparation

In der Phase Data Preparation werden die gesammelten Daten bereinigt, sodass sie für den Anwendungsfall entsprechend vorbereitet sind (Sarkar et al., 2018). Die Vorbereitung der Daten ist die zeitintensivste Phase und benötigt zwischen 60% - 70% der Zeit. Es sollte hier auf keinen Fall an Zeit gespart werden, da die Qualität der Daten über die Performance des Modells entscheiden.

2.14.4. Modeling

In der vierten Phase werden nun die vorbereiteten Daten verwendet, um das Modell zu trainieren (Sarkar et al., 2018). Hier ist der Prozess iterativ mit der Modell Evaluation. Das Modell wird trainiert, die Evaluation durchgeführt und aufgrund der erhaltenen Erkenntnisse wird das Modell weiter verbessert. Die Idee besteht darin, aus den verschiedenen Modellen dasjenige zu finden, welches die Anforderungen des Anwendungsfalls am besten erfüllt.

2.14.5. Testing und Evaluation

In der Evaluations Phase wird überprüft, wie gut das Modell sich für den Anwendungsfall eignet (Tierney, 2014). Zur Überprüfung des Modells werden verschiedene Metriken verwendet.

Folgende zentrale Punkte werden bewertet (Sarkar et al., 2018):

- Die erarbeiteten Modelle werden bezogen auf den Anwendungsfall mittels entsprechenden Metriken bewertet und eine Reihenfolge definiert.
- Schwachstellen im gesamten Prozess werden dokumentiert, um zukünftig Fehler zu vermeiden.
- Annahmen oder Einschränkungen, welche vor dem Modelling getroffen wurden, werden nun überprüft und gegebenenfalls widerlegt.

2.14.6. Deployment

In der letzten Phase des CRISP-DM Modell wird definiert, wie die Modelle in der entsprechenden Umgebung implementiert werden können (Tierney, 2014). Die Umsetzung kann in unterschiedlichster Form erfolgen, von der Unterstützung in Entscheidungsprozessen bis hin zu einer automatisierten Integration in operativen Systemen. Hierbei steht immer im Zentrum einen Mehrwert für das Unternehmen oder den jeweiligen Anwendungsfall zu schaffen.

3. Methoden und Vorgehen

In diesem Abschnitt wird die methodische Vorgehensweite zur Beantwortung der Forschungsfrage vorgestellt. Im Fokus steht das Fine Tuning eines öffentlich verfügbaren NER-Modells, das personenbezogene Daten klassifiziert. Ziel des Fine Tunings ist es, das Modell mit neuen Kategorien besonders schützenswerter Daten zu erweitern. Die Umsetzung der Arbeit orientiert sich am zuvor beschriebenen CRISP-DM Modell.

3.1. Technische Informationen

Der Code wurde in der Programmiersprache Python geschrieben. Die Implementierung wurde in Jupyter Notebook auf einem MacBook Pro mit Apple M3 Max Chip durchgeführt. Das Fine Tuning des Modells wurde lokal vorgenommen. Es wurden verschiedene Bibliotheken verwendet, darunter Transformers, Datasets und Scikit-Learn. Diese sind in den im Anhang verfügbaren Jupyter Notebooks definiert und dokumentiert.

3.2. Verwendung generativer KI

Für die Umsetzung der technischen Bestandteile der Arbeit kam die Version 4 von ChatGPT der Firma OpenAI zum Einsatz (OpenAI, 2024). Die Verwendung bezog sich auf die Erstellung des Codes. Zur sprachlichen Glättung einzelner Abschnitte wurde ChatGPT ebenfalls verwendet. Die Vorschläge wurden gezielt genutzt, um die Lesbarkeit zu erhöhen. Der fachliche Gehalt wurde dabei nicht verändert. Die zentralen Begriffe und die inhaltlichen Aussagen wurden nochmals überprüft und bei Bedarf manuell angepasst.

3.3. Datenerhebung

Um das NER-Modell gezielt auf die Erkennung zusätzlicher Entitäten weiter zu trainieren, wurden passende Trainingsdaten benötigt.

Ein Teil der Daten wurde über die Plattform Hugging Face bezogen. Hugging Face ist eine Community-basierte Plattform für Machine Learning, auf welcher verschiedene Datensätze und vortrainierte Modelle veröffentlicht werden. Besonders bekannt ist die

Plattform für seine Vielzahl an Transformer Modellen, die vor allem bei NLP-Aufgaben zum Einsatz kommen.

Die Daten für das Training setzten sich aus zwei Komponenten zusammen. Als erste Komponente kam ein Datensatz von Hugging Face zum Einsatz, der bereits gelabelte personenbezogene Daten enthielt. Dieser wurde von AI4Privacy veröffentlicht und ist ausdrücklich für akademische Zwecke freigegeben (pii-masking-400k - HUG, 2024). Ein Vorteil dieses Datensatzes bestand darin, dass das ausgewählte NER-Modell die darin enthaltenen Entitäten bereits erkennen konnte und somit eine gute Grundlage für das Fine Tuning bildete.

Da der AI4Privacy-Datensatz allerdings nicht alle Labels enthielt, welche für das Ziel-Modell benötigt wurden, wurde ein neuer Datensatz erstellt. Die Daten wurden mithilfe eines Template-Ansatzes generiert und bestehen aus synthetischen Beispielen. Dabei wurde sichergestellt, dass alle Labels enthalten sind, die vom Modell neu erlernt werden sollten.

Bei beiden Datensätzen wurde sorgfältig darauf geachtet, dass keine Datenschutzverletzungen vorliegen. Sämtliche Daten sind vollständig synthetisch und beinhalten keine realen personenbezogenen Informationen.

3.4. Business Understanding

Das Verständnis der Datenschutzgrundverordnung (DSGVO) der EU und des revidierten Schweizer Datenschutzgesetzes (DSG) bildet die Grundlage für die technische Umsetzung. In den Gesetzen ist definiert, welche Informationen als personenbezogen oder besonders schützenswert gelten. Die zu klassifizierenden Datenkategorien wurden auf Basis der Gesetzte festgelegt. Diese dienten als Grundlage, um eine präzise Erkennung zu ermöglichen und um sicherzustellen, dass das Modell rechtlich relevante Daten korrekt erkennt.

3.5. Data Understanding

Die Datengrundlage bildete der ai4privacy/pii-masking-400k Datensatz von Hugging Face. Dieser enthielt bereits gelabelte Daten für insgesamt 17 Entitäten und eignete sich somit optimal als Ausgangspunkt um das gewählte Modell weiterzuentwickeln. Ergänzend dazu wurden zusätzliche synthetische Daten erstellt, damit das Modell neue Labels erlernen konnte.

Das Verständnis der Daten wurde durch die systematische Exploration erlangt. Dabei wurden die Spalten und deren Strukturen des AI4Privacy-Datensatzes analysiert. Es wurden die relevanten Felder für das Training vorgemerkt. Potenzielle Herausforderungen wurden erkannt, um sie in der Datenvorbereitung gezielt zu adressieren.

Auch die Verteilung der bestehenden Labels wurde untersucht, um ein besseres Verständnis über deren Verteilung zu erlangen. Für den eigens generierten Datensatz wurden vorgängig die zusätzlichen Entitätsklassen definiert. Es wurde sich hierbei auf 3 zusätzliche Labels fokussiert. Der erstellte Datensatz wurde ebenfalls deskriptiv analysiert.

3.6. Data Preparation

In der Phase der Data Preparation wurde der AI4Privacy-Datensatz umfassend bereinigt. Der Datensatz war bereits in Train- und Test-Splits aufgeteilt, dieser wurde wieder zusammengeführt, da die Verteilung nach der Bereinigung der Sprache und des Zusammenführens mit dem generierten Datensatzes keine passende Verteilung mehr ausgewiesen hätte. Spalten, welche nicht notwendig waren, wurden direkt bereinigt.

Da der AI4Privacy-Datensatz mehrere Sprachen enthielt, aber die ergänzenden Daten auf Deutsch sein sollten, wurde dieser ebenfalls auf die deutschsprachigen Texte begrenzt. Die Labels waren im AI4Privacy-Datensatz noch nicht in der Form, die das Modell benötigt hätte. Die Daten wurden daher entsprechend aufbereitet und in neuen Spalten abgelegt, um eine geeignete Basis für das Training und die Weiterverarbeitung zu schaffen.

Die synthetischen Daten wurden bewusst in Deutsch erstellt. Die Daten wurden mittels Templates generiert und mit den gewünschten Werten ausgefüllt. Es wurden Sammlungen von Wörtern erstellt und diese in die passenden Templates gefüllt. Diese Sammlungen waren unterteilt in die Labels, welche das Modell lernen sollte. Ebenfalls wurden je nach Satzstruktur des Templates eine Sammlung für Nomen und eine Sammlung für Adjektive erstellt. Die für das Training notwendige Label-Zuweisung erfolgte im gleichen Zug mit dem Erstellen der Daten. Die Daten wurden mittels Python Code erstellt. Nach der Erstellung der Daten wurden diese mit dem AI4Privacy-Datensatz zusammengeführt.

Anschliessend wurden die Daten aufgeteilt in einen 80/20-Split. Von den 20% wurden 10% dem Test-Split zugewiesen und 10% dem Validation-Split. Um im Modell zu definieren, welche Labels es lernen soll, wurde eine Label2Id-Zuweisung gemacht. Anschliessend wurden die Texte mit dem entsprechenden Tokenizer des Modells tokenisiert. Das Modell, welches gewählt wurde, ist das piiranha-v1-detect-personal-information Modell. Dieses wurde, wie auch der AI4Privacy-Datensatz auf Hugging Face zur Verfügung gestellt. Das Modell wurde bereits auf dem AI4Privacy-Datensatz trainiert und war somit sehr gut geeignet für eine Weiterentwicklung.

3.7. Modelling

Nachdem die Tokenisierung erfolgt ist, wurde die Evaluationsfunktion definiert, welche aus Precision, Recall und F1-Score bestand. Der ganze Prozess des Fine Tunings wurde mit der Transformers Bibliothek umgesetzt. Nach der Definition der Evaluationsfunktion wurden die Training Arguments gewählt, welche eine Vielzahl an Trainingsparametern Verfügung stellt. Anschliessend wurde das Modell trainiert.

3.8. Evaluation

Das trainierte Modell wurde mit den Testdaten bewertet. Dabei kamen etablierte Metriken wie Precision, Recall und der F1-Score zum Einsatz, die im Bereich der Named Entity Recognition als Standard gelten. Dabei wurde die Modellleistung auf Ebene der einzelnen Labels analysiert. Aufgrund der Erkenntnisse aus der Evaluation wurden verschiedene Massnahmen definiert und es wurde wieder in eine der vorherigen Phasen gewechselt, um das Modell weiter zu verbessern.

3.9. Deployment

Eine tatsächliche Integration des Modells wurde im Rahmen dieser Arbeit nicht umgesetzt. Stattdessen werden mögliche Anwendungsfälle beschrieben, um potenzielle Einsatzszenarien aufzuzeigen.

4. Empirischer Teil

Es wurde untersucht, wie sich Personendaten mittels Named Entity Recognition erkennen lassen. Zwei Aspekte standen hierbei besonders im Fokus: die Daten und das Modell. Auf der einen Seite die Daten, welche für ein Fine Tuning notwendig sind, zum anderen das Modell, welches verwendet werden kann, um die domänenspezifische Anpassung vorzunehmen.

Nachfolgend wird das genaue Vorgehen von der Datenaufbereitung über das Fine Tuning bis zur Evaluation beschrieben. Ziel war es, ein praxisnahes Modell zu entwickeln, das in der Lage ist, personenbezogene und besonders schützenswerte Daten zuverlässig zu klassifizieren.

4.1. Deskriptive Analyse des AI4Privacy-Datensatzes

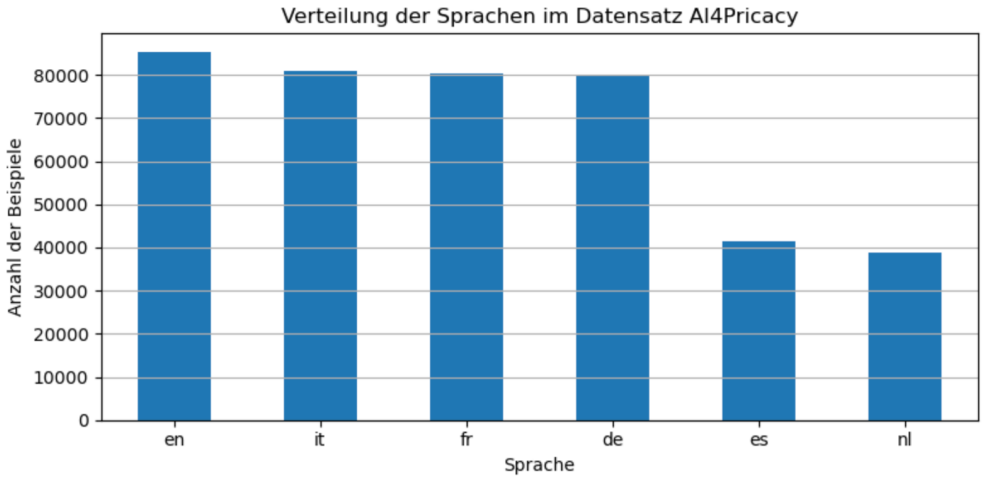
Die Daten für das Fine Tuning des Modells wurden auf Hugging Face gefunden. Der Datensatz besteht aus 9 Spalten und 406,896 Zeilen. Darin enthalten sind verschiedene Texte, welche mit Personally Identifiable Information (PII) versetzt wurden. Es wurden 17 Klassen an PII-Daten verwendet. Die verschiedenen Klassen werden in der nachfolgenden Tabelle beschrieben.

Tabelle 3: Übersicht der AI4Privacy Entitäten

Labelname	**Name**	**Beispiele aus dem Datensatz**
Accountnum	Kontonummer	02999894672953515995973 7041443
Buildingnum	Hausnummer	19118 167
City	Stadt	Berlin Churwalden
Creditcardnumber	Kreditkartennummer	3123628353294143250 675964923641
Dateofbirth	Geburtsdatum	May/58 8th January 1999
Driverlicensenum	Führerscheinnummer	D768740898123 253162622
Email	E-Mail-Adresse	faozzsd379223@outlook.com 52siddharta@aol.com
Givenname	Vorname	Kalaivanan Seita
Idcardnum	Identitätskartennummer	2149769228 GQ88644CL
Password	Passwort	.&yM#jEi\0s^ 15|Xm~w+iMBC
Socialnum	Sozialversicherungsnummer	777-80-2312 756.9684.5139.05
Street	Strasse	Welxander Strasse Mülibachstrasse
Surname	Nachname	van der Boog Dolderer
Telephonenum	Telefonnummer	+18 93-764 5097 078 8968568
Username	Benutzername	faozzsd379223 bahara.cathers19
Zipcode	Postleitzahl	LS29 96813

Im Datensatz sind 6 verschiedene Sprachen enthalten: Englisch, Italienisch, Französisch, Niederländisch, Spanisch und Deutsch. Die Sprachen im Datensatz sind wie folgt verteilt:

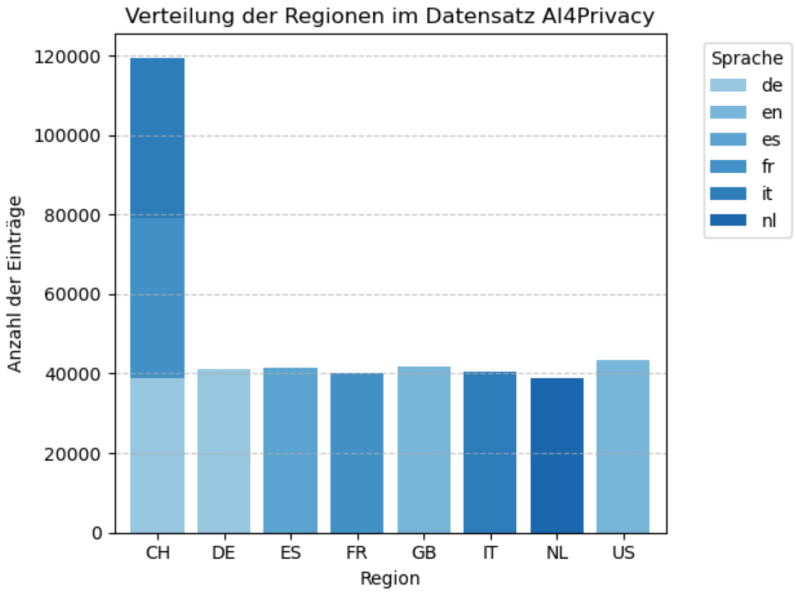
Abbildung 8: Verteilung der Sprachen im Datensatz AI4Privacy



Da das Modell auf Deutsch weitertrainiert werden soll, fokussiert sich die Auswertung auf die deutschsprachigen Einträge. Von den insgesamt 406’896 Einträgen sind 79’880 deutschsprachige Einträge vorhanden. Damit zählt Deutsch zusammen mit Englisch, Italienisch und Französisch zu den Hauptsprachen im Datensatz.

Neben der sprachlichen Zuordnung erfolgte auch eine regionale Einteilung der Datensätze. Dabei wurden Länder wie das Vereinigte Königreich, die Vereinigten Staaten, Italien, Frankreich, die Schweiz, die Niederlande, Deutschland und Spanien berücksichtigt.

Abbildung 9: Verteilung der Regionen im Datensatz AI4Privacy



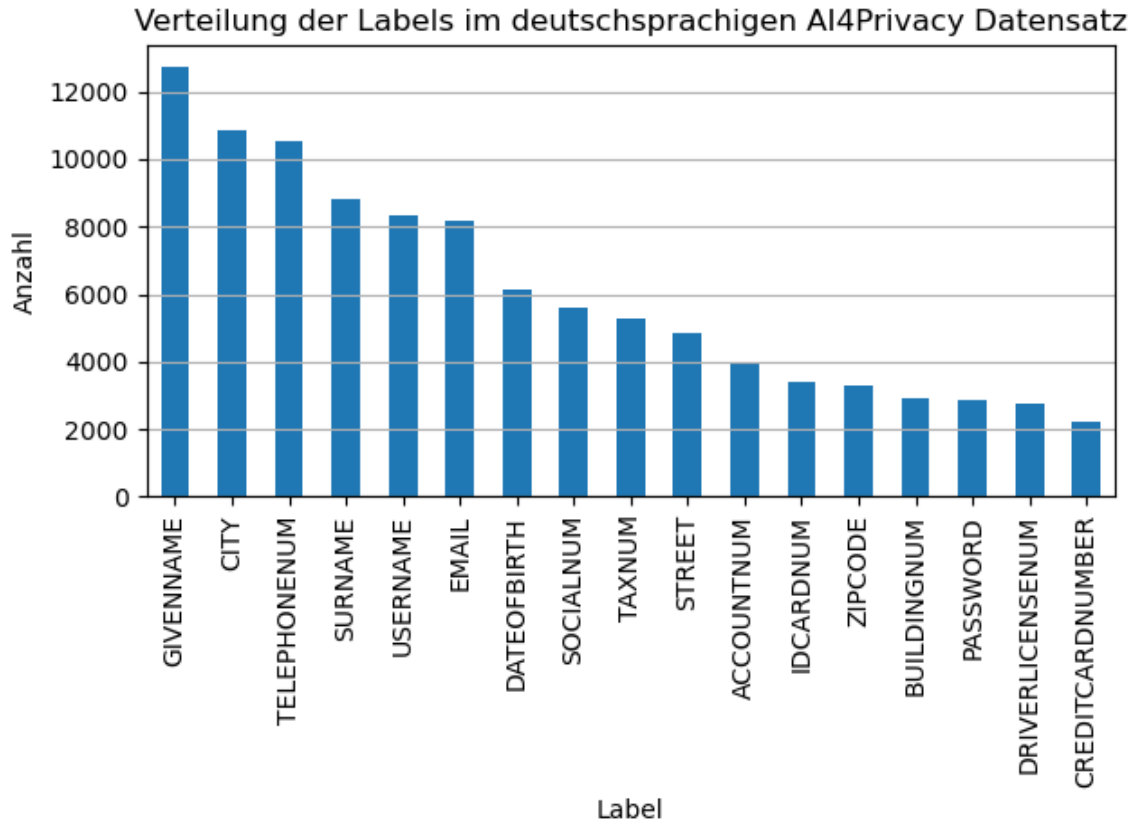
Die deutschsprachigen Daten verteilen sich zu etwa gleichen Teilen auf die Schweiz und Deutschland. Die regionale Zuordnung ist relevant, da sich Datenformate wie beispielsweise die Sozialversicherungsnummer länderspezifisch unterscheiden.

Tabelle 4: Unterschiede der Sozialversicherungsnummern verschiedener Länder

Land	**Sozialversicherungsnummer**
Vereinigtes Königreich (GB)	QQ 12 34 56 C
Vereinigte Staaten (US)	123-45-6789
Italien (IT)	RSSMRA85M01H501Z
Frankreich (FR)	1 84 12 75 123 456 89
Schweiz (CH)	756.1234.5678.97
Niederlande (NL)	123456782
Deutschland (DE)	65 130892 M 123
Spanien (ES)	12345678Z

Nach der sprachlichen Filterung ergibt sich im deutschsprachigen Datensatz folgende Verteilung der Labels:

Abbildung 10: Verteilung der Labels im deutschsprachigen AI4Privacy-Datensatz



Die am wenigsten vertretene PII-Klasse ist die Kreditkartennummer mit 2’219 Einträgen. Die am häufigsten vertretene Klasse der Vorname mit 12’718 Einträgen. Der Durchschnittswert aller Klassen liegt bei 6’036 Beispielen.

Um die Struktur der Einträge besser nachvollziehen zu können, folgt ein Ausschnitt aus dem AI4Privacy-Datensatz. Dieser bildet die Basis für die anschliessende Beschreibung des Datenstruktur.

Abbildung 11: Ausschnitt aus dem AI4Privacy-Datensatz

	source_text	locale	language	split	privacy_mask	uid	masked_text	mbert_tokens	mbert_token_classes
0	Für die Teilnahme an den Studentenvertretungswahlen. Benutzername: 55kawa.moongamackal, Telephon: +41.408-185 6375, Adresse: Grands Monts 7.2, PLZ: 2400, BIC: PAXKUSFB, ID-Nummer: T1496246, Passwort: o-Q0, Sozialversicherungsnummer: 756.0558.5784.07.	CH	de	train	[{'label': 'USERNAME', 'start': 67, 'end': 86, 'value': '55kawa.moongamackal', 'label_index': 1}, {'label': 'TELEPHONENUM', 'start': 98, 'end': 114, 'value': '+41.408-185 6375', 'label_index': 1}, {'label': 'STREET', 'start': 125, 'end': 137, 'value': 'Grands Monts', 'label_index': 1}, {'label': 'BUILDINGNUM', 'start': 138, 'end': 141, 'value': '7.2', 'label_index': 1}, {'label': 'ZIPCODE', 'start': 148, 'end': 152, 'value': '2400', 'label_index': 1}, {'label': 'IDCARDNUM', 'start': 180, 'end': 188, 'value': 'T1496246', 'label_index': 1}, {'label': 'PASSWORD', 'start': 200, 'end': 204, 'value': 'o-Q0', 'label_index': 1}, {'label': 'SOCIALNUM', 'start': 233, 'end': 249, 'value': '756.0558.5784.07', 'label_index': 1}]	145002	Für die Teilnahme an den Studentenvertretungswahlen. Benutzername: [USERNAME_1], Telephon: [TELEPHONENUM_1], Adresse: [STREET_1] [BUILDINGNUM_1], PLZ: [ZIPCODE_1], BIC: PAXKUSFB, ID-Nummer: [IDCARDNUM_1], Passwort: [PASSWORD_1], Sozialversicherungsnummer: [SOCIALNUM_1].	[Für, die, Teilnahme, an, den, Studenten, ##vert, ##retung, ##swahlen, ., Ben, ##utz, ##erna, ##me, :, 55, ##kawa, ., moon, ##gama, ##cka, ##l, ,, Tel, ##ep, ##hon, :, +, 41, ., 408, -, 185, 637, ##5, ,, Ad, ##resse, :, Grands, Monts, 7, ., 2, ,, PL, ##Z, :, 2400, ,, B, ##IC, :, PA, ##X, ##K, ##US, ##F, ##B, ,, ID, –, Nummer, :, T1, ##49, ##6, ##24, ##6, ,, Pass, ##wort, :, o, -, Q, ##0, ,, So, ##zial, ##versi, ##cherung, ##s, ##nummer, :, 756, ., 055, ##8, ., 578, ##4, ., 07, .]	[O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, B-USERNAME, I-USERNAME, I-USERNAME, I-USERNAME, I-USERNAME, I-USERNAME, I-USERNAME, O, O, O, O, O, B-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, O, O, O, O, B-STREET, I-STREET, B-BUILDINGNUM, I-BUILDINGNUM, I-BUILDINGNUM, O, O, O, O, B-ZIPCODE, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, B-IDCARDNUM, I-IDCARDNUM, I-IDCARDNUM, I-IDCARDNUM, I-IDCARDNUM, O, O, O, O, O, B-PASSWORD, I-PASSWORD, I-PASSWORD, O, O, O, O, O, O, O, O, O, B-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, O]

Zur besseren Übersicht sind die Spalten des Datensatzes in der folgenden Tabelle zusammengefasst:

Tabelle 5: Übersicht der Spalten des AI4Privacy-Datensatzes

Spalte	**Beschriftung**
source_text	Der ursprüngliche, unmaskierte Text mit potenziellen PII-Daten.
locale	Region, aus der der Text stammt.
language	Sprache des Textes.
split	Kennzeichnung, ob es sich um Trainings- oder Validierungsdaten handelt.
privacy_mask	Strukturierte Angaben zu erkannten PII-Labels: Name, Position und Wert.
uid	Eindeutige Kennung des Eintrags zur gezielten Referenzierung.
masked_text	Der Text mit Platzhaltern für erkannte PII-Daten.

mbert_tokens	Tokenisierte Version des source_text mithilfe des Multilingual BERT Tokenizers.
mbert_token_classes	BIO-Tags, die den jeweiligen Tokens zugeordnet sind.

In der Spalte ganz links, „source_text“, befindet sich der eigentliche Text, welcher PII-Daten enthält. In der Spalte „locale“ wird die Region deklariert, aus der der Text kommt, während in der Spalte „language“ die Sprache enthalten ist. Die Spalte „split“ enthält entweder „train“ oder „validation“, diese Verteilung wurde bereits von AI4Privacy vorgenommen.

Die „privacy_mask“ enthält strukturierte Informationen über die Labels: den Labelnamen, den Start- und Endpunkt sowie auch den Wert des Labels. Die Spalte „uid“ enthält eine einzigartige Nummer des Eintrags um auf diesen entsprechend Referenzieren zu können.

Die Spalte „masked_text“ enthält den Text mit Platzhaltern für die PII-Informationen. „mbert_tokens“ enthält den Text mittels Multilingual BERT Tokenizers in Tokens aufgeteilt. In der Spalte „mbert_token_classes“ sind die BIO-Tags zu den dazugehörigen Tokens aus „mbert_tokens“ enthalten.

Die vorangegangene Analyse des AI4Privacy-Datensatzes hat gezeigt, dass eine grosse Anzahl gelabelter und strukturierter Beispiele in deutscher Sprache vorhanden sind. Um diese Daten sinnvoll für das Fine Tuning nutzen zu können, ist die Auswahl eines passenden Modells zentral. Das Modell sollte bereits in der Lage sein, personenbezogene Daten zu erkennen, und zugleich genügend flexibel sein, um auf neue Labels angepasst werden zu können. Im folgenden Abschnitt wird das gewählte Modell vorgestellt und die Entscheidungsgrundlage erläutert.

4.2. Wahl des Modells für das Fine Tuning

Das gewählte Modell wurde, wie der Datensatz von AI4Privacy, ebenfalls auf Hugging Face publiziert. Das iiiorg/piiranha-v1-detect-personal-information Modell wurde auf den AI4Privacy Daten trainiert, weshalb das Modell die 17 verschiedenen PII-Daten aus dem Datensatz bereits erkennen kann. Dies ermöglicht ein Fine Tuning auf

ergänzende Labels, um eine spezifische Erweiterung auf besonders schützenswerte Daten vorzunehmen.

Der Vorteil des vortrainierten Modells liegt darin, dass eine grosse Menge an Ressourcen (Zeit für das Training und Geld für die Hardware) eingespart werden konnten. Eine Anschaffung einer solchen Grafikkarte wäre im Rahmen dieser Arbeit nicht möglich gewesen. Da das Modell eine sehr gute Ausgangslage bietet, wurde aufgrund der begrenzten Rechenressourcen darauf verzichtet, ein Modell vollständig neu zu trainieren. Die Wahl des Modells schränkt zugleich die Auswahl der Libraries ein, wodurch die Umsetzung mithilfe der Transformers-Bibliothek stattfand.

Abbildung 12: Precision, Recall und F1-Score des Piiranha-Modells für die im AI4Privacy-Datensatz enthaltenen Entitäten

Entity	Precision	Recall	F1-Score	Support
ACCOUNTNUM	0.84	0.87	0.85	3575
BUILDINGNUM	0.92	0.90	0.91	3252
CITY	0.95	0.97	0.96	7270
CREDITCARDNUMBER	0.94	0.96	0.95	2308
DATEOFBIRTH	0.93	0.85	0.89	3389
DRIVERLICENSENUM	0.96	0.96	0.96	2244
EMAIL	1.00	1.00	1.00	6892
GIVENNAME	0.87	0.93	0.90	12150
IDCARDNUM	0.89	0.94	0.91	3700
PASSWORD	0.98	0.98	0.98	2387
SOCIALNUM	0.93	0.94	0.93	2709
STREET	0.97	0.95	0.96	3331
SURNAME	0.89	0.78	0.83	8267
TAXNUM	0.97	0.89	0.93	2322
TELEPHONENUM	0.99	1.00	0.99	5039
USERNAME	0.98	0.98	0.98	7680
ZIPCODE	0.94	0.97	0.95	3191
micro avg	0.93	0.93	0.93	79706
macro avg	0.94	0.93	0.93	79706
weighted avg	0.93	0.93	0.93	79706

*Quelle: (*piiranha-v1 - HUG*, 2025)*

In der Tabelle sind die Precision-, Recall- und F1-Scores des Piiranha-Modells für die im AI4Privacy-Datensatz enthaltenen Entitäten dargestellt. Die Ergebnisse zeigen, dass das Modell bereits eine hohe Ausgangsgenauigkeit bei der Erkennung personenbezogener Informationen wie Benutzernamen, Telefonnummern oder Adressen erreicht. Der durchschnittliche F1-Score (weighted avg) liegt bei 0.93, was eine solide Basis für die Weiterentwicklung darstellt. Bei der Nutzung rein deutschsprachiger Datensätze kann es zu leichten Abweichungen kommen, da das Modell auf einem multilingualen Korpus trainiert und evaluiert wurde.

Obwohl das Modell bereits eine starke Performance bei klassischen PII-Entitäten aufweist, deckt es die besonders schützenswerten Datenkategorien gemäss Artikel 9 DSGVO noch nicht ab. Im nächsten Schritt wurden daher die AI4Privacy Daten, die die Grundlage für das Fine Tuning bilden, gezielt für den Einsatz mit dem Piiranha-Modell aufbereitet. Ergänzend zu den AI4Privacy-Daten wurden synthetische Daten zur Erweiterung der Labels erstellt. Eine detaillierte Beschreibung des Vorgehens zur Erstellung der synthetischen Daten findet sich in Kapitel Synthetische Daten erstellen.

4.3. Datenvorbereitung des AI4Privacy-Datensatzes

Nachdem der Datensatz deskriptiv untersucht wurde, folgt nun die Bereinigung und somit die Vorbereitung auf das Fine Tuning des Modells. Für diesen Schritt werden nicht alle Spalten des AI4Privacy-Datensatzes benötigt. Der Datensatz wurde auf deutschsprachige Einträge eingeschränkt. Die Beschränkung auf die deutsche Sprache wurde bewusst gewählt, da auch die zusätzlichen, synthetisch erzeugten Beispiele nur auf Deutsch formuliert wurden. Die Entscheidung für die deutsche Sprache ergab sich aus den vorhandenen Sprachkenntnissen und dem besseren Verständnis für die in dieser Region üblichen Datenstrukturen. Zudem sollte so eine möglichst konsistente Grundlage für das spätere Fine Tuning geschaffen werden.

Der Input für das spätere Fine Tuning benötigt den Text und die dazugehörigen Labels. Die Spalten „locale“, „language“, „split“, „privacy_mask“, „uid“, „masked_text“ wurden somit nicht benötigt. Für das Fine Tuning des Modells reichen die Spalten „mbert_tokens“ und „mbert_token_classes“ allerdings nicht aus, da jedes Modell seine eigene Tokenisierung durchführt. Im konkreten Fall unterscheidet sich die

Tokenisierung des Piiranha-Modells grundlegend von derjenigen des ursprünglich verwendeten BERT-Modells. Daher müssen die Tokens und zugehörigen Labels passend zur Tokenisierung des Piiranha-Modells neu ausgerichtet werden.

In der folgenden Abbildung ist ein Satz dargestellt, der einmal mit dem Piiranha-Tokenizer und einmal mit dem bert-base-cased-Tokenizer tokenisiert wurde.

Abbildung 13: Vergleich zwei verschiedener Tokenizer

```python
from transformers import AutoTokenizer

text = "Die Teilnahme an Max's Konfirmation ist für die Eltern verpflichtend."

# Tokenizer
piiranha_tokenizer = AutoTokenizer.from_pretrained("iiiorg/piiranha-v1-detect-personal-information")
bert_tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")

# Tokenisierung mit Piiranha
piiranha_tokens = piiranha_tokenizer.tokenize(text)

# Tokenisierung mit BERT
bert_tokens = bert_tokenizer.tokenize(text)

print("Tokenisierung mit Piiranha Tokenizer:")
print(piiranha_tokens)

print("\nTokenisierung mit BERT Tokenizer:")
print(bert_tokens)
```

```
Tokenisierung mit Piiranha Tokenizer:
['_Die', '_T', 'eilnahme', '_an', '_Max', "'", 's', '_K', 'onfirmation', '_ist', '_für', '_die', '_Eltern', '_', 'verpflicht', 'end', '.']

Tokenisierung mit BERT Tokenizer:
['Die', 'Te', '##il', '##nah', '##me', 'an', 'Max', "'", 's', 'Ko', '##n', '##fi', '##rma', '##tion', 'is', '##t', 'für', 'die', 'El', '##ter
n', 've', '##rp', '##f', '##lich', '##tend', '.']
```

Beim Piiranha-Tokenizer wird der Beginn eines neuen Wortbestandteils durch einen Unterstrich „_“ markiert. Alle folgenden Subwords ohne Unterstrich gehören zum gleichen Wort. Beispiel: Das Wort „Teilnahme“ wird in die Tokens „_T“ und „eilnahme“ aufgeteilt. Der Unterstrich bei „_T“ zeigt den Beginn des Wortes an, „eilnahme“ ergänzt es.

Im Gegensatz dazu kennzeichnet der bert-base-cased-Tokenizer Subwords mit einem vorangestellten „##“, wodurch ersichtlich wird, dass es sich um Fortsetzungen eines zuvor begonnenen Wortes handelt. Beispiel: Das Wort „Teilnahme“ wird in „Te“, „##il“, „##nah“ und „##me“ zerlegt. Nur das erste Token steht ohne Markierung, alle weiteren Subwords beginnen mit „##“.

Nicht nur die Tokenisierung unterscheidet sich zwischen dem Piiranha-Modell und den mBERT-Tokens aus dem AI4Privacy-Datensatz. Die Art des Taggings ist ebenfalls

unterschiedlich (mbert_token_classes Spalte). Während bei BERT das klassische BIO-Tagging verwendet wird, nutzt Piiranha nur I- und O-Tags.

Das Tagging stellt den Zusammenhang zwischen den Tokens und ihren zugehörigen Entitätslabels her. Beim BIO-Tagging steht B für den Beginn einer Entität, I für Tokens innerhalb einer Entität und O für Tokens, die keiner Entität zugeordnet sind. Beim I/O-Tagging, wie es beim Piiranha-Modell verwendet wird, entfällt die separate Markierung des Entitätsbeginns. Tokens, die Teil einer Entität sind, werden direkt mit dem entsprechenden I-Label versehen. Das Modell muss sich somit nicht darum kümmern, explizit den Start einer Entität zu erkennen, sondern nur, ob ein Token zu einer Entität gehört oder nicht.

Abbildung 14: Mapping von Tokens zu IO-Tags

_Max	'	s	_K	onfirmation	_ist	_he	ute	.
I-GIVENNAME	O	O	I-REL	I-REL	O	O	O	O

Die Abbildung zeigt die Tokenisierung des Satzes „Max's Konfirmation ist heute." mit dem Piiranha-Tokenizer. Dabei steht „_Max" für den Wortanfang und trägt das Label I-GIVENNAME, „_K" sowie „onfirmation" erhalten das Label I-REL, ohne dass, wie im BIO-Tagging üblich, zwischen B- und I- unterschieden wird. Die restlichen Tokens wie „_ist" und „_heute" sind mit O gekennzeichnet.

Zum Vergleich zeigt die folgende Abbildung einen Auszug aus dem ursprünglichen AI4Privacy-Trainingsdatensatz. Auf der linken Seite ist der Ausgangstext dargestellt, in der Mitte die Tokenisierung mit dem mBERT-Tokenizer und rechts die zugehörigen Entitätslabels im BIO-Format. Dadurch werden die Unterschiede zur IO-Struktur des Piiranha-Modells nochmals deutlich.

Abbildung 15: mbert-Tokens aus dem AI4Privacy-Datensatz

source_text	mbert_tokens	mbert_token_classes
Für die Teilnahme an den Studentenvertretungswahlen. Benutzername: 55kawa.moongamackal, Telephon: +41.408-185 6375, Adresse: Grands Monts 7.2, PLZ: 2400, BIC: PAXKUSFB, ID-Nummer: T1496246, Passwort: o-Q0, Sozialversicherungsnummer: 756.0558.5784.07.	[Für, die, Teilnahme, an, den, Studenten, ##vert, ##retung, ##swahlen, ., Ben, ##utz, ##erna, ##me, :, 55, ##kawa, ., moon, ##gama, ##cka, ##l, ,, Tel, ##ep, ##hon, :, +, 41, ., 408, -, 185, 637, ##5, ,, Ad, ##resse, :, Grands, Monts, 7, ., 2, ,, PL, ##Z, :, 2400, ,, B, ##IC, :, PA, ##X, ##K, ##US, ##F, ##B, ,, ID, -, Nummer, :, T1, ##49, ##6, ##24, ##6, ,, Pass, ##wort, :, o, -, Q, ##0, ,, So, ##zial, ##versi, ##cherung, ##s, ##nummer, :, 756, ., 055, ##8, ., 578, ##4, ., 07, .]	[O, O, O, O, O, O, O, O, O, O, O, O, O, O, B-USERNAME, I-USERNAME, I-USERNAME, I-USERNAME, I-USERNAME, I-USERNAME, I-USERNAME, O, O, O, O, O, B-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, O, O, O, O, B-STREET, I-STREET, B-BUILDINGNUM, I-BUILDINGNUM, I-BUILDINGNUM, O, O, O, O, B-ZIPCODE, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, B-IDCARDNUM, I-IDCARDNUM, I-IDCARDNUM, I-IDCARDNUM, I-IDCARDNUM, O, O, O, O, B-PASSWORD, I-PASSWORD, I-PASSWORD, I-PASSWORD, O, O, O, O, O, O, O, O, B-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, O]

Für die Weiterverwendung im Fine Tuning musste das ursprüngliche BIO-Tagging an das vom Piiranha-Modell verwendete IO-Format angepasst werden. Dazu wurden die B-Tags entfernt und alle Tokens, die zu einer Entität gehören, direkt mit I-Tags versehen.

Abbildung 16: Angepasste mBERT-Tokens auf IO-Tags

mbert_tokens	io_tags
[Für, die, Teilnahme, an, den, Studenten, ##vert, ##retung, ##swahlen, ., Ben, ##utz, ##erna, ##me, :, 55, ##kawa, ., moon, ##gama, ##cka, ##l, ,, Tel, ##ep, ##hon, :, +, 41, ., 408, -, 185, 637, ##5, ,, Ad, ##resse, :, Grands, Monts, 7, ., 2, ,, PL, ##Z, :, 2400, ,, B, ##IC, :, PA, ##X, ##K, ##US, ##F, ##B, ,, ID, -, Nummer, :, T1, ##49, ##6, ##24, ##6, ,, Pass, ##wort, :, o, -, Q, ##0, ,, So, ##zial, ##versi, ##cherung, ##s, ##nummer, :, 756, ., 055, ##8, ., 578, ##4, ., 07, .]	[O, O, O, O, O, O, O, O, O, O, O, O, O, O, I-USERNAME, I-USERNAME, I-USERNAME, I-USERNAME, I-USERNAME, I-USERNAME, I-USERNAME, O, O, O, O, O, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, I-TELEPHONENUM, O, O, O, O, I-STREET, I-STREET, I-BUILDINGNUM, I-BUILDINGNUM, I-BUILDINGNUM, O, O, O, O, I-ZIPCODE, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, I-IDCARDNUM, I-IDCARDNUM, I-IDCARDNUM, I-IDCARDNUM, I-IDCARDNUM, O, O, O, O, I-PASSWORD, I-PASSWORD, I-PASSWORD, I-PASSWORD, O, O, O, O, O, O, O, O, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, I-SOCIALNUM, O]

Anschliessend wurden die einzelnen Tokens wieder zu vollständigen Wörtern zusammengeführt. Nachdem diese Anpassung erfolgt ist, wurden die Labels auf die einzelnen Wörter gemapped, sodass die neue Tokenisierung erfolgen kann. Durch diese Anpassung konnte beim Tokenisieren der Parameter (is_split_into_words=True) verwendet werden, um eine korrekte Zuordnung zwischen Wörtern und ihren jeweiligen Labels sicherzustellen.

Abbildung 17: Fertiges Mapping von Wörtern auf IO-Tags

words	word_io_tags
[Für, die, Teilnahme, an, den, Studentenvertretungswahlen., Benutzername:, 55kawa.moongamackal,, Telephon:, +41.408-185, 6375,, Adresse:, Grands, Monts, 7.2,, PLZ:, 2400,, BIC:, PAXKUSFB,, ID-Nummer:, T1496246,, Passwort:, o-Q0,, Sozialversicherungsnummer:, 756.0558.5784.07.]	[O, O, O, O, O, O, O, I-USERNAME, O, I-TELEPHONENUM, I-TELEPHONENUM, O, I-STREET, I-STREET, I-BUILDINGNUM, O, I-ZIPCODE, O, O, O, I-IDCARDNUM, O, I-PASSWORD, O, I-SOCIALNUM]

Die Abbildung zeigt einen Ausschnitt aus dem vorbereiteten Trainingsdatensatz. In der ersten Spalte sind die vorverarbeiteten Wörter des Originaltexts kommagetrennt aufgelistet. Die zweite Spalte enthält die zugehörigen Labels im IO-Format, wobei jedes Label exakt auf die Wortunterteilungen aus der ersten Spalte abgestimmt ist. Beim Mapping dieser Labels trat vereinzelt das Problem auf, dass nachfolgende Satzzeichen übernommen wurden, wie beim letzten Label I-SOCIALNUM ersichtlich ist. In diesem Fall ist der Punkt Teil der erkannten Entität geworden. Auf eine vertiefte Anpassung des Codes wurde verzichtet, da dies im angedachten Anwendungsfall, wie etwa dem Schwärzen von Dokumenten, keine wesentliche Rolle spielt. Es ist in diesem Kontext unkritisch, wenn auch Satzzeichen mitgeschwärzt werden. Aus diesem Grund wurde die Anpassung in dieser Form belassen.

Diese Anpassungen waren notwendig, da ein Modell beim Fine Tuning exakt dieselbe Struktur und Tokenisierung benötigt, wie sie später auch bei der Verarbeitung neuer Eingabedaten verwendet wird. Eine konsistente Tokenisierung stellt sicher, dass das Modell die Eingaben korrekt interpretiert und die erlernten Muster zuverlässig anwenden kann.

4.4. Wahl der neuen Labels

Nachdem die Basis mit einem geeigneten Modell sowie einem vorbereiteten Datensatz gelegt wurde, folgt nun die Erweiterung des Labelsets um besonders schützenswerte Personendaten. Im Piiranha-Modell lag der Fokus primär auf personenidentifizierenden Informationen. Für diese Arbeit sollte das Modell jedoch auch besonders schützenswerte Daten erkennen. Laut DSGVO (Art. 9) zählen dazu unter anderem Informationen zur rassischen und ethnischen Herkunft, politischen Meinung, religiösen oder weltanschaulichen Überzeugung, Gewerkschaftszugehörigkeit, genetischen und biometrischen Daten, Gesundheitsdaten sowie Daten zur sexuellen Orientierung.

Der Fokus wurde dabei auf drei dieser sensiblen Kategorien gelegt: Religion (REL), ethnische Herkunft (ETHN) und sexuelle Orientierung (SOR). Im Rahmen des Fine Tunings wurde das bestehende Labelset gezielt um diese drei Klassen erweitert, um die Erkennung sensibler personenbezogener Daten im Sinne der Datenschutzkonformität zu verbessern.

Die Themen Religion, ethnische Herkunft und sexuelle Orientierung konnten eindeutig abgegrenzt und anhand typischer Beispiele wie „Christentum“, „afrikanische Herkunft“ oder „homosexuelle Orientierung“ für Anwendungsfälle im Datenschutzkontext modelliert werden. Bei der Auswahl der neuen Labels wurde darauf geachtet, dass sowohl eine hohe rechtliche Relevanz als auch ein fundiertes Domänenverständnis vorliegt.

Auf weitere Kategorien wie politische Meinungen oder Gesundheitsdaten wurde im ersten Schritt verzichtet, da hier eine noch tiefere fachliche Auseinandersetzung notwendig gewesen wäre, um eine präzise Abbildung sicherzustellen.

Mit der Erweiterung um diese drei Labels wurde eine ausgewogene Grundlage geschaffen, um das Modell für zentrale Aufgaben der Datenschutzkonformität in Unternehmen weiterzuentwickeln.

4.5. Synthetische Daten erstellen

Nachdem die neuen Labels definiert wurden, stellte sich die Frage, wie diese effektiv in das Fine Tuning eingebunden werden können. Da im ursprünglichen Datensatz keine ausreichende Anzahl an Beispielen für besonders schützenswerte Kategorien wie Religion, ethnische Herkunft oder sexuelle Orientierung vorhanden war, wurde entschieden, gezielt synthetische Trainingsdaten zu erzeugen. Ziel war es, dem Modell aussagekräftige Beispiele für diese Entitäten bereitzustellen, um eine zuverlässige Klassifikation zu ermöglichen.

Da es sich bei der Umsetzung um eine Supervised Learning Methode handelt, wurden im Rahmen dieser Arbeit gelabelte synthetische Daten erstellt. Die Datensätze

enthalten unter anderem die drei neu definierten Labels, welche das Modell zusätzlich erlernen soll.

Zur Erstellung eines spezialisierten Datensatzes wurde ein Template-basierter Ansatz gewählt. Zunächst wurde geprüft, ob eine Generierung der Trainingsdaten ausschliesslich mittels Chat GPT möglich ist. Dabei zeigte sich jedoch, dass die erstellten Sätze häufig sprachliche Unstimmigkeiten aufwiesen und die Daten nur begrenzt variabel waren. Zudem konnte die zuverlässige Zuweisung der Labels nicht sichergestellt werden, was eine manuelle Nachbearbeitung in grossem Umfang nötig gemacht hätte.

Aus diesen Gründen wurde entschieden, einen kontrollierten Template-Ansatz zu verfolgen. Dieser bringt zwar ein gewisses Risiko für Overfitting mit sich, gewährleistet jedoch, dass die erzeugten Daten sprachlich korrekt gelabelt sind.

Es wurden Templates erstellt, die mit passenden Begriffen befüllt wurden. Die verwendeten Wörter wurden dabei in semantische Kategorien wie Nomen und Adjektive unterteilt, um eine natürliche Satzstruktur sicherzustellen. Pro Label-Kategorie wurden mehrere Sätze je Wortart definiert. Die Wortlisten wurden teils manuell zusammengestellt, teils mithilfe eines generativen Sprachmodells erstellt. Letzteres erforderte eine aufwändige Nachbearbeitung, da viele der automatisch generierten Begriffe unpassend oder unvollständig waren. Anschliessend wurden die Templates mit den bereinigten Begriffen befüllt. Die Auswahl der Templates und Wörter erfolgte dabei zufällig. Die resultierenden Sätze wurden, analog zur Aufbereitung des AI4Privacy-Datensatzes, in Wörter unterteilt und mit den entsprechenden IO-Labels versehen.

Die nachfolgende Abbildung zeigt eine Auswahl an Templates, die im Rahmen der Datengenerierung für das Label ETHN verwendet wurden. Platzhalter wie {givenname}, {ethn_adj} oder {username} wurden später zufällig mit passenden Begriffen aus vorbereiteten Listen befüllt.

Abbildung 18: Beispiel einer Liste an Templates für das Label ETHN

```
ethn_adj_temp = [
                "{givenname} sieht {ethn_adj} aus, von sich sind viele Bilder auf seinem Instagramm account {username}.",
                "Mit seinen dunkenbraunen Haaren und den blauen Augen sieht er typisch {ethn_adj} aus.",
                "Er sieht {ethn_adj} aus, mit kurzen, blonden Haaren, die ordentlich frisiert sind, und hellblauen Augen,
                die einen ruhigen, nachdenklichen Ausdruck haben.
                Sein Gesicht ist markant, mit hohen Wangenknochen und einer leicht kantigen Kinnpartie.",
                "Lebenslauf Name: {givenname} {surname}, Geburtsdatum: {dateofbirth}, Staatsangehörigkeit: {ethn_adj}"
]
```

Ein Auszug aus einer solchen Wörterliste ist in der nächsten Abbildung dargestellt. Die Begriffe bestehen überwiegend aus Adjektiven, die ethnische oder kulturelle Zugehörigkeiten ausdrücken.

Abbildung 19: Beispiel einer Wörterliste bestehend aus Adjektiven für das Label EHTN

```
ethn_adj = [
            "deutsch", "türkisch", "polnisch", "russisch", "italienisch", "slawisch", "britisch",
            "französisch", "portugiesisch", "nordisch", "keltisch", "germanisch", "balkanisch",
            "mazedonisch", "serbisch", "albanisch", "israelisch", "armenisch", "finnisch", "ungarisch",
            "slowakisch", "ukrainisch", "katalanisch", "bosnisch", "bulgarisch", "schwäbisch", "griechisch",
            "schweizer", "türkisch-deutsch"
]
```

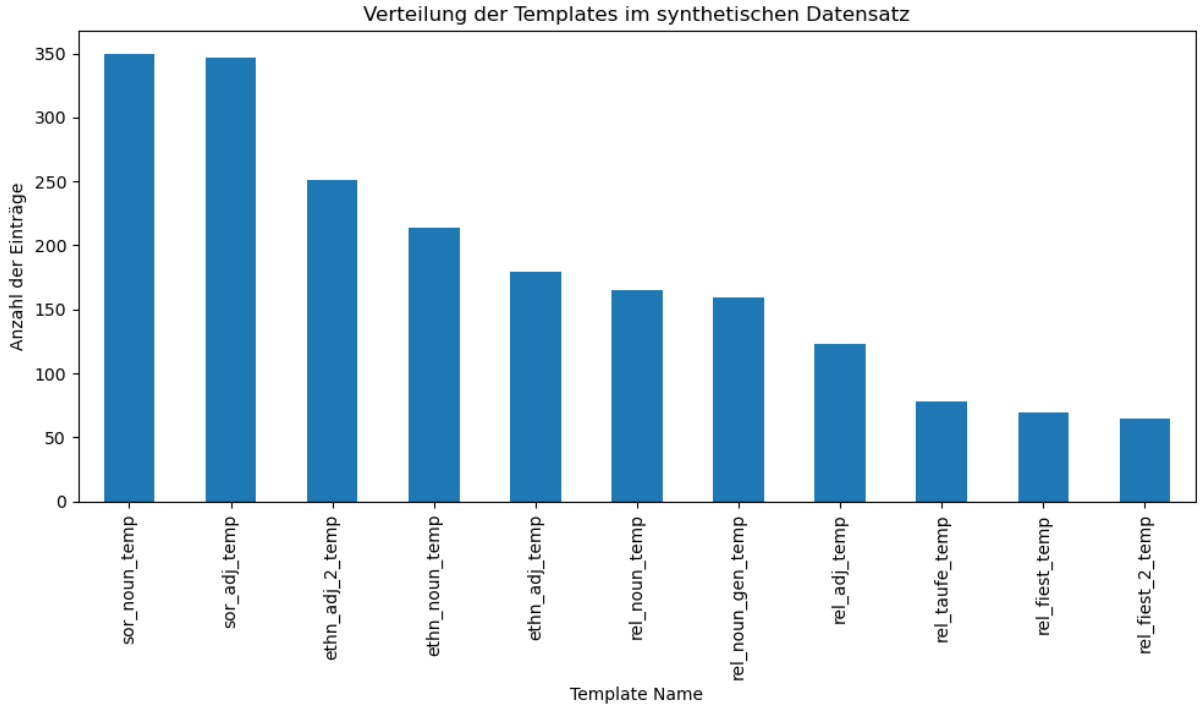
Zusätzlich wurde ein Label-Mapping erstellt, sodass die Daten korrekt verarbeitet und als konsistenter Input für das Fine Tuning genutzt werden können. Die folgende Abbildung zeigt, wie die generierten Sätze in Einzelelemente zerlegt und mit den entsprechenden IO-Labels versehen wurden. Neben dem Rohtext enthält der Datensatz die verwendete Template-Kategorie.

Abbildung 20: Ausschnitt des synthetischen Datensatzes

synt_text	synt_words	synt_words_labels	template_name
Auf der Pride in Essen war Mirella mit einer Flagge unterwegs, die seine aromantische Identität zeigte.	[Auf, der, Pride, in, Essen, war, Mirella, mit, einer, Flagge, unterwegs,, die, seine, aromantische, Identität, zeigte.]	[O, O, O, I-ETHN, I-CITY, O, I-GIVENNAME, O, O, O, O, O, O, O, O, O]	sor_noun_temp
Folgende Personen haben Interesse an einer Beratung bezüglich ihrer sexuellen Orientierung: Chavez, Geb.: 26.09.20, Orientierung: straight, lebt in: Berlin, Chavez, Geb.: 26.09.20, Orientierung: straight, lebt in: Berlin, Chavez, Geb.: 26.09.20, Orientierung: straight, lebt in: Berlin	[Folgende, Personen, haben, Interesse, an, einer, Beratung, bezüglich, ihrer, sexuellen, Orientierung:, Chavez,, Geb.:, 26.09.20,, Orientierung:, straight,, lebt, in:, Berlin,, Chavez,, Geb.:, 26.09.20,, Orientierung:, straight,, lebt, in:, Berlin,, Chavez,, Geb.:, 26.09.20,, Orientierung:, straight,, lebt, in:, Berlin]	[O, O, O, O, O, O, O, O, O, O, O, I-SURNAME, O, I-DATEOFBIRTH, O, I-SOR, O, O, I-CITY, I-SURNAME, O, I-DATEOFBIRTH, O, I-SOR, O, O, I-CITY, I-SURNAME, O, I-DATEOFBIRTH, O, I-SOR, O, O, I-CITY]	sor_noun_temp
Am Holi versammelt sich die Gemeinde, um den spirituellen Akt des Gebets und der Feier zu teilen.	[Am, Holi, versammelt, sich, die, Gemeinde,, um, den, spirituellen, Akt, des, Gebets, und, der, Feier, zu, teilen.]	[O, I-REL, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O]	sor_noun_temp

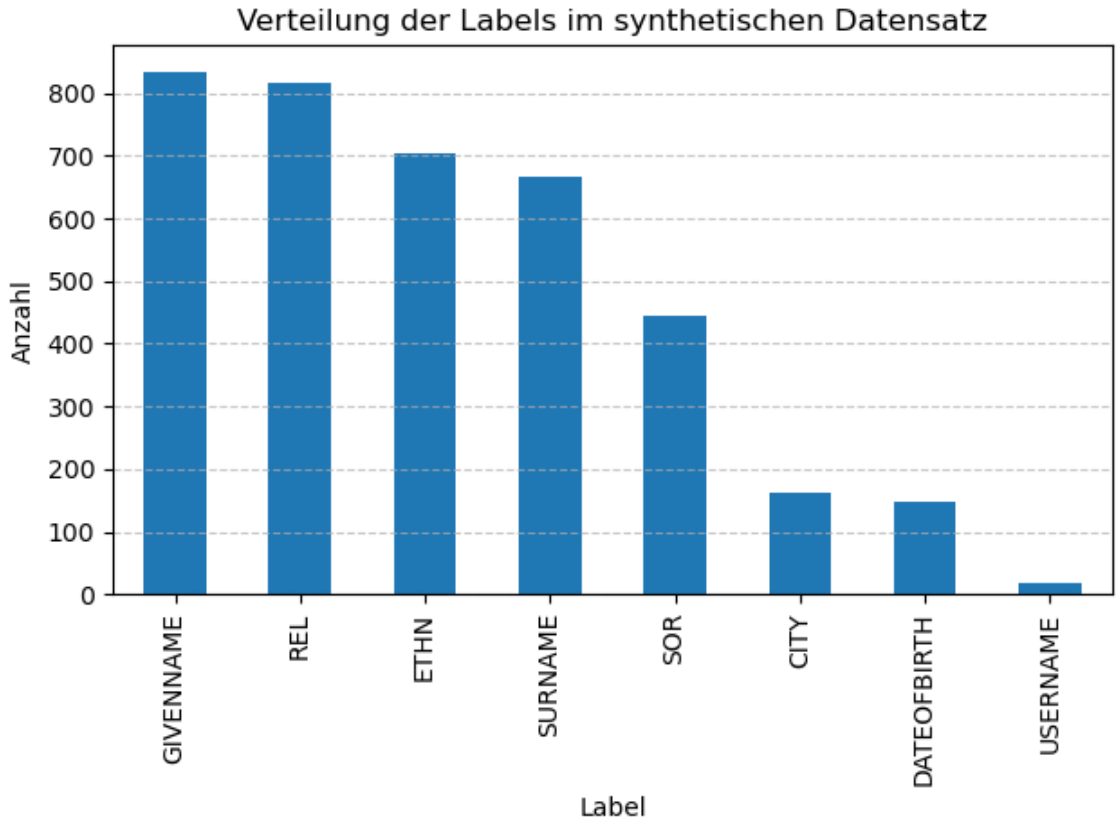
Die Verteilung der verwendeten Templates wurde dokumentiert, um einen strukturierten Überblick über die Zusammensetzung des synthetischen Datensatzes zu gewährleisten. Die folgende Abbildung zeigt, wie häufig einzelne Templates im finalen Datensatz verwendet wurden. Die Anzahl der verwendeten Satz-Templates unterscheidet sich nur geringfügig zwischen den Kategorien. Für die Labels REL, ETHN und SOR wurden jeweils 38 bis 39 verschiedene Vorlagen erstellt. Die Anzahl generierter Sätze pro Label-Kategorie ist ausgeglichen. Für das Label REL wurden 659 Sätze erstellt, für ETHN 644 und für SOR 697. Die in der Abbildung sichtbaren Unterschiede ergeben sich daraus, dass einzelne Template-Kategorien innerhalb einer der besonders schützenswerten Datenkategorie häufiger verwendet wurden als andere.

Abbildung 21: Verteilung der Templates im synthetischen Datensatz



Neben der Verteilung der Template-Kategorien wurde auch die Häufigkeit, der im synthetischen Datensatz enthaltenen Labels, analysiert. Die folgende Abbildung zeigt, wie oft jedes Label insgesamt vorkommt.

Abbildung 22: Verteilung der Labels im synthetischen Datensatz



Die besonders schützenswerten Kategorien REL, ETHN und SOR wurden gezielt in ausreichender Anzahl eingebunden, damit das Modell neue Entitäten verlässlich erlernen kann. Weitere Labels wie GIVENNAME, SURNAME, DATEOFBIRTH, CITY und USERNAME wurden ergänzt, um die Sätze sprachlich zu vervollständigen und inhaltlich realistischer zu gestalten. Diese tragen zur Satzqualität bei, stehen jedoch nicht im Zentrum der eigentlichen Modellanpassung.

Eine Einschränkung dieses Ansatzes besteht darin, dass durch die Verwendung fixer Templates eine gewisse Reduktion der Sprachvariabilität entsteht, was sich auf die spätere Generaliserungsfähigkeit des Modells auswirken kann.

Eine Einschränkung dieses Ansatzes besteht darin, dass durch die Verwendung fixer Templates die sprachliche Variabilität eingeschränkt wird. Dies kann die Generalisierungsfähigkeit des Modells in realen Anwendungsfällen potenziell begrenzen. Gleichzeitig bietet der strukturierte Aufbau den Vorteil, dass die erzeugten Daten kontrolliert erstellt und korrekt gelabelt sind.

Die so erzeugten synthetischen Daten bilden somit zusammen mit dem AI4Privacy-Datensatz die Grundlage für das Fine Tuning des Modells.

4.6. Fine Tuning

Im Folgenden wird das technische Vorgehen beim Fine Tuning beschrieben, von der Datenkonsolidierung über die Tokenisierung bis hin zur Trainings- und Evaluationsphase.

Für das Fine Tuning wurden zwei Datenquellen verwendet: die synthetisch generierten Daten als dataset_synthetic.csv sowie der öffentlich verfügbare Datensatz ai4privacy/pii-masking-400k, der zuvor als dataset_ai4privacy.csv exportiert und wie im Kapitel Datenvorbereitung des AI4Privacy-Datensatzes beschrieben bereinigt wurde.

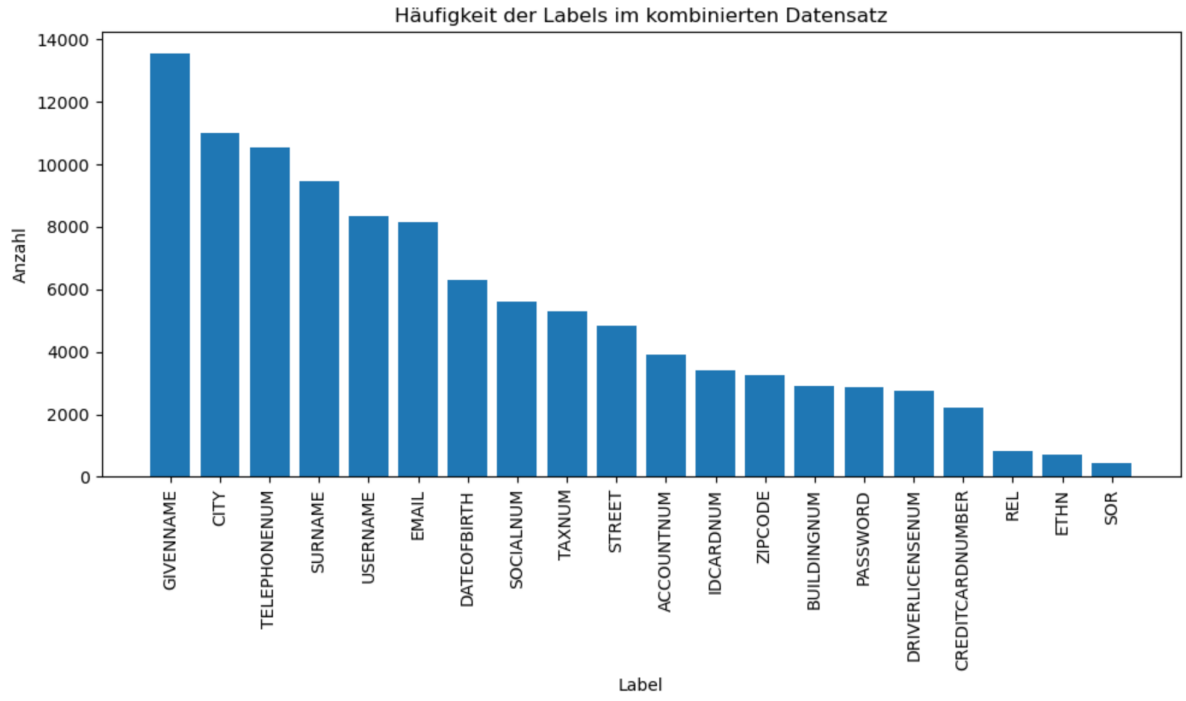
Beide Quellen wurden importiert und zusammengeführt. Dabei wurden die Felder words und word_io_tags aus dem dataset_ai4privacy.csv sowie synt_words und synt_words_labels aus dem dataset_synthetic.csv verwendet. Anschliessend wurden die Spaltennamen harmonisiert, sodass beide Datensätze in einem konsolidierten DataFrame unter den Spaltennamen tokens und label_names zusammengeführt werden konnten.

Die Daten des ursprünglichen AI4Privacy-Datensatzes wurden bewusst im Training belassen, um Catastrophic Forgetting zu vermeiden. Dieses Phänomen tritt auf, wenn ein Modell durch ausschliessliches Training auf neuen Daten zuvor erlernte Informationen verliert. Um dem entgegenzuwirken, wurde das Konzept des Replays angewendet. Dabei werden bekannte Trainingsbeispiele weiterhin einbezogen, sodass bereits gelernte Entitäten erhalten bleiben und im besten Fall sogar gestärkt werden.

Nach der Zusammenführung beider Quellen ergibt sich ein konsolidierter Datensatz, der sowohl bestehende als auch neu eingeführte Entitäten enthält. Die Gesamtverteilung aller Labels ist in der folgenden Abbildung dargestellt. Die neu eingeführten Labels REL, ETHN und SOR sind im Vergleich zu den übrigen Labels

weniger häufig vertreten. Diese Verteilung spiegelt eine realistische Datensituation wider, da besonders schützenswerte personenbezogene Informationen in realen Textquellen typischerweise seltener auftreten.

Abbildung 23: Verteilung der Labels im konsolidierten Datensatz



Im nächsten Schritt wurde ein Label-Mapping erstellt, das alle vorhandenen sowie neu eingeführten Label Kategorien in ein konsistentes Format überführte. Die zusätzlichen Entitäten REL, ETHN und SOR wurden dabei ebenfalls in das Mapping aufgenommen.

Abbildung 24: Label-ID-Zuweisung

```python
# Feste Label-Zuweisung der alten und auch neuen Labels
label2id = {
    'I-ACCOUNTNUM': 0,
    'I-BUILDINGNUM': 1,
    'I-CITY': 2,
    'I-CREDITCARDNUMBER': 3,
    'I-DATEOFBIRTH': 4,
    'I-DRIVERLICENSENUM': 5,
    'I-EMAIL': 6,
    'I-GIVENNAME': 7,
    'I-IDCARDNUM': 8,
    'I-PASSWORD': 9,
    'I-SOCIALNUM': 10,
    'I-STREET': 11,
    'I-SURNAME': 12,
    'I-TAXNUM': 13,
    'I-TELEPHONENUM': 14,
    'I-USERNAME': 15,
    'I-ZIPCODE': 16,
    'O': 17,
    'I-REL': 18,
    'I-ETHN': 19,
    'I-SOR': 20
}
id2label = {i: label for label, i in label2id.items()}
```

Im nächsten Schritt wurde das Modell iiiorg/piiranha-v1-detect-personal-information gemeinsam mit dem zugehörigen Tokenizer geladen. Die Konfiguration wurde so angepasst, dass alle im Label-Mapping definierten Klassen übernommen wurden. Dafür wurde das label2id-Mapping an die Modellkonfiguration übergeben. Zusätzlich wurde ignore_mismatched_sizes=True gesetzt, um die Klassifikationsschicht an die geänderte Anzahl Labels anzupassen. Dieser Parameter sorgt dafür, dass beim Laden eines vortrainierten Modells die letzte Schicht, also die für die Klassifizierung zuständige Ebene, neu initialisiert wird, da die Anzahl der Zielklassen im eigenen Datensatz nicht mehr mit der des ursprünglichen Modells übereinstimmt. In diesem Fall war das notwendig, da neue Entitäten wie REL, ETHN und SOR ergänzt wurden.

Abbildung 25: Piiranha-Modell laden

```
# Tokenizer und Modell laden
model_name = "iiiorg/piiranha-v1-detect-personal-information"
tokenizer = AutoTokenizer.from_pretrained(model_name)

config = AutoConfig.from_pretrained(model_name, num_labels=len(label2id))
config.label2id = label2id
config.id2label = id2label

model = AutoModelForTokenClassification.from_pretrained(
    model_name,
    config=config,
    ignore_mismatched_sizes=True  # letzte Schicht neu initialisiert
)
```

Im Anschluss wurde die Funktion convert_row_to_features implementiert, um die Eingabedaten für das Modell vorzubereiten. Dabei erfolgte die Tokenisierung word-piece-basiert unter Verwendung des Parameters is_split_into_words=True, da die Eingabetexte bereits in einzelne Wörter unterteilt vorlagen. Diese Einstellung stellte sicher, dass die ursprünglichen Wortgrenzen bei der Tokenisierung berücksichtigt wurden. Die Word-Piece-Methode ermöglichte es dem Modell, auch seltene oder unbekannte Begriffe durch Subtokenisierung korrekt zu verarbeiten.

Um die ursprünglichen Wort-Labels korrekt auf die Tokenstruktur zu übertragen, wurde die Methode word_ids() eingesetzt. Sie ordnet jedem erzeugten Subtoken das passende Ursprungswort zu. In der Funktion wurden anschliessend die IO-Labels entsprechend diesen Zuordnungen ausgerichtet. Die tokenisierten Eingabedaten, bestehend aus input_ids, attention_mask und den zugehörigen Labels, wurden abschliessend in ein einheitliches Format überführt und als DatasetDict gespeichert.

Abbildung 26: Aufbereitung der Daten für das Fine Tuning

```python
# Daten konvertieren
def convert_row_to_features(row):
    tokens = row["tokens"]
    labels = row["label_names"]

    encoding = tokenizer(
        tokens,
        is_split_into_words=True,
        return_offsets_mapping=True,
        truncation=True,
        padding="max_length",
        max_length=128
    )

    word_ids = encoding.word_ids()
    aligned_labels = []
    for word_id in word_ids:
        if word_id is None or word_id >= len(labels):
            aligned_labels.append(-100)
        else:
            aligned_labels.append(label2id.get(labels[word_id], label2id["O"]))

    return {
        "input_ids": encoding["input_ids"],
        "attention_mask": encoding["attention_mask"],
        "labels": aligned_labels
    }

dataset_dicts = [convert_row_to_features(row) for _, row in df_combined.iterrows()]
dataset = Dataset.from_list(dataset_dicts)
```

Nach der Umwandlung aller Einträge in das gewünschte Format wurden die Daten für das Training vorbereitet. Dazu erfolgte eine Aufteilung in Trainings-, Validierungs- und Testdaten im Verhältnis 80 % zu 10 % zu 10 %. Ziel war es, eine möglichst gleichmässige Verteilung der Klassen über alle Teilmengen hinweg sicherzustellen.

Abbildung 27: Train-/Test-/Validation-Split

```python
# Split in Train/Val/Test
split = dataset.train_test_split(test_size=0.2, seed=42)
val_test = split["test"].train_test_split(test_size=0.5, seed=42)
dataset_final = DatasetDict({
    "train": split["train"],
    "validation": val_test["train"],
    "test": val_test["test"]
})
```

Im Anschluss an die Datenaufteilung wurde das Training vorbereitet. Dabei kam ein Data Collator zum Einsatz, der sicherstellt, dass Tokens und die zugehörigen Labelinformationen in korrekter Form und Länge in Batches verarbeitet werden. Zur Bewertung der Modellleistung wurde die Bibliothek seqeval verwendet. Die

Kennzahlen Precision, Recall und F1-Score wurden über eine eigene compute_metrics-Funktion berechnet, die nur die relevanten Token berücksichtigt. Dabei werden alle Positionen mit dem Wert -100, also Padding oder ignorierte Tokens, von der Auswertung ausgeschlossen.

Abbildung 28: Funktion zur Modellbewertung

```
def compute_metrics(pred):
    true_labels = pred.label_ids
    pred_labels = np.argmax(pred.predictions, axis=2)

    true_tags, pred_tags = [], []

    for true_seq, pred_seq in zip(true_labels, pred_labels):
        true_seq_labels = []
        pred_seq_labels = []
        for t, p in zip(true_seq, pred_seq):
            if t != -100:
                true_seq_labels.append(id2label[t])
                pred_seq_labels.append(id2label[p])
        true_tags.append(true_seq_labels)
        pred_tags.append(pred_seq_labels)

    return {
        "precision": precision_score(true_tags, pred_tags),
        "recall": recall_score(true_tags, pred_tags),
        "f1": f1_score(true_tags, pred_tags)
    }
```

Das Training selbst wurde mit dem Trainer-Modul aus der Hugging Face Transformers-Bibliothek durchgeführt. Die Trainingsparameter umfassten drei Epochen, eine Batchgrösse von acht sowie eine Evaluation nach jeder Epoche. Die geringe Anzahl an Epochen wurde bewusst gewählt, um einem möglichen Overfitting entgegenzuwirken. Die Batchgrösse wurde mit acht bewusst niedrig gewählt, um Memory Engpässen vorzubeugen und ein stabiles Training mit moderaten Ressourcen zu ermöglichen.

Abbildung 29: Definition der Fine Tuning Parameter

```
training_args = TrainingArguments(
    output_dir="./piiranha-model-finetuned",
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=3,
    evaluation_strategy="epoch",
    save_strategy="epoch",
    logging_steps=100,
    report_to="none",
    push_to_hub=True,
    hub_model_id="HuggingLil/pii-sensitive-ner-german",
    hub_strategy="end"
)
```

Das Modell wurde mit den zuvor definierten Parametern trainiert und nach jeder Epoche automatisch evaluiert. Grundlage dafür bildete die compute_metrics-Funktion, die im Trainer eingebunden wurde und sicherstellt, dass nach jeder Epoche die Metriken Precision, Recall und F1-Score berechnet und protokolliert werden. Die finale Bewertung des Modells erfolgte nach Abschluss des Trainings auf dem separaten Testdatensatz. Die Trainingsdauer im finalen Versuch betrug nur wenige Stunden.

Abbildung 30: Fine Tuning des Modells

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=dataset_final["train"],
    eval_dataset=dataset_final["validation"],
    tokenizer=tokenizer,
    data_collator=DataCollatorForTokenClassification(tokenizer),
    compute_metrics=compute_metrics
)

trainer.train()
```

Nach Abschluss des Fine Tunings wurde das trainierte Modell einer systematischen Evaluierung unterzogen. Ziel war es, die Leistung hinsichtlich der Erkennung sowohl bestehender als auch neu eingeführter Labels zu bewerten und die Qualität der Vorhersagen nachvollziehbar einzuordnen. Im folgenden Kapitel werden die dabei erzielten Ergebnisse sowie beispielhafte Klassifikationen vorgestellt und analysiert.

5. Evaluation und Ergebnisse

In diesem Kapitel wird die Leistung des Modells analysiert. Der Fokus liegt dabei auf der Frage, wie zuverlässig das Modell sowohl bestehende als auch neu eingeführte Entitäten erkennen kann. Zunächst werden die verwendeten Bewertungsmethoden beschrieben und die Ergebnisse auf dem Testdatensatz vorgestellt. Anschliessend erfolgt eine nähere Betrachtung einzelner Klassifikationen sowie eine qualitative Einordnung der Modellleistung.

5.1. Evaluierung der Modellleistung

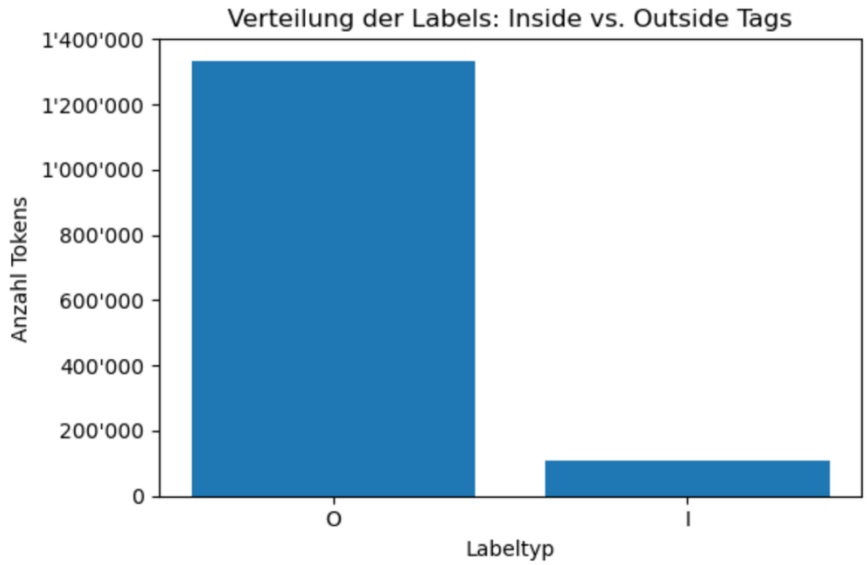
Ziel der Evaluierung ist es, zu überprüfen, wie gut das Modell in der Lage ist, sowohl bestehende als auch die neuen Entitäten korrekt zu erkennen. Die Bewertung erfolgt auf dem Testset, das nach der Konsolidierung der Daten erstellt wurde.

Das Modell wurde im Verlauf der Arbeit mehrfach trainiert. Die synthetischen Daten schrittweise erweitert und inhaltlich angepasst, um eine bessere Abdeckung der neuen Labels (REL, ETHN, SOR) zu erreichen. Die dargestellten Ergebnisse und Klassifikationen beziehen sich auf den finalen Stand des Modells nach dem abgeschlossenen Fine Tuning.

Zur Bewertung der Modellleistung wurden Precision, Recall und F1-Score verwendet. Auf die Accuracy wurde bewusst verzichtet, da der Datensatz im Fall von Named Entity Recognition eine unausgeglichene Verteilung zwischen Entitäten und Nicht-Entitäten aufweist. Der Grund dafür ist, dass in NER-Aufgaben meist nur einzelne Wörter innerhalb eines Textes zu einer Entität gehören, während der Grossteil der Wörter keiner Entität zugeordnet ist.

Wie in der nachfolgenden Abbildung ersichtlich, ist die Verteilung zwischen Outside- und Inside-Tags, wie bei NER üblich, sehr ungleich.

Abbildung 31: Verteilung der Labels: Inside- vs. Outside-Tags



92.61 % der Labels gehören zu den Outside-Tags, lediglich 7.39 % zu den Inside-Tags. Eine hohe Accuracy wäre in diesem Fall nicht aussagekräftig, da ein Modell auch dann einen hohen Wert erreichen könnte, wenn es nahezu ausschliesslich Outside-Tags vorhersagt. Würde das Modell keine einzige Entität erkennen, läge die Accuracy trotzdem bereits bei 92.61 % und wäre somit irreführend.

Die Evaluierung der Modellleistung lässt sich in zwei Teile gliedern und erfolgt zum einen während des Trainings und zum anderen nach Abschluss des Trainings.

Während des Trainings wurde nach jeder Epoche eine Auswertung auf dem Validierungsdatensatz durchgeführt, um die Entwicklung des Modells zu verfolgen. Dafür kam die Funktion compute_metrics-Funktion zum Einsatz (Abbildung 28: Funktion zur Modellbewertung). Diese wurde dem Trainer beim Start des Trainings übergeben, wodurch die Berechnung der Kennzahlen automatisch nach jeder Epoche erfolgte.

Nach dem Training wurde ergänzend eine Auswertung auf dem Testdatensatz durchgeführt. Dabei machte das Modell Vorhersagen auf Basis der Daten im Testdatensatz. Anschliessend wurde geprüft, ob diese mit den tatsächlichen Labels übereinstimmen. Die Bewertung erfolgte sowohl über alle Einträge hinweg als auch bezogen auf jedes einzelne Label.

Die Evaluierung bildet die Grundlage, um die Leistung des finalen Modells einzuordnen. Im nächsten Abschnitt werden die Klassifikationsergebnisse pro Label sowie Beispielsätze gezeigt, in denen das Piiranha-Modell mit dem feinjustierten Modell verglichen wird.

5.2. Einordnung des Trainingsprozesses

Die folgende Abbildung zeigt den Verlauf des Trainingsprozesses über drei Epochen hinweg. Aufgeführt sind jeweils der Training Loss, der Validation Loss sowie die Metriken Precision, Recall und F1-Score für jede Epoche.

Abbildung 32: Trainingsverlauf über drei Epochen

Epoch	Training Loss	Validation Loss	Precision	Recall	F1
1	0.056000	0.042775	0.891075	0.906620	0.898780
2	0.022700	0.036899	0.911758	0.927868	0.919743
3	0.013300	0.034462	0.923580	0.931223	0.927386

Bereits nach der ersten Epoche wurde ein F1-Score von 0.8988 auf dem Validierungsset erreicht. Dieser hohe Anfangswert lässt sich unter anderem damit erklären, dass das Modell bereits vortrainiert war und die ursprünglichen Labels aus dem AI4Privacy-Datensatz kannte. In der zweiten Epoche stieg dieser Wert auf 0.9197 an und erreichte in der dritten Epoche 0.9274. Diese Entwicklung zeigt, dass das Modell über die drei Trainingsphasen hinweg konsistente Fortschritte machte.

Der Training Loss nahm mit jeder Epoche weiter ab, was darauf hinweist, dass das Modell die Trainingsdaten immer besser verarbeiten konnte. Gleichzeitig verbesserte sich auch der Validation Loss kontinuierlich, was ein Zeichen dafür ist, dass sich das Modell nicht nur auf die Trainingsdaten spezialisiert, sondern seine Vorhersagen auch auf neuen, ungesehenen Daten verbessert hat. Die Differenz zwischen Training und Validation Loss blieb über alle Epochen hinweg gering, was auf ein stabiles Lernverhalten ohne Anzeichen von Overfitting hindeutet.

Die ersten Ergebnisse zeigen eine stabile Entwicklung und deuten auf ein vielversprechendes Modellverhalten hin. Im nächsten Kapitel werden die Klassifikationsergebnisse des Modells sowohl quantitativ als auch qualitativ analysiert.

5.3. Klassifikationsergebnisse

Zur Bewertung der Modellleistung wurden zunächst die aggregierten Klassifikationsmetriken über alle Entitäten hinweg betrachtet. Die folgende Tabelle zeigt die Gesamtperformance des feinjustierten Modells, gemessen an Precision, Recall und F1-Score.

Abbildung 33: Übersicht über die Gesamtperformance des Modells

	precision	recall	f1-score	support
accuracy	0.993	0.993	0.993	0.993
macro avg	0.963	0.959	0.961	363703.000
weighted avg	0.993	0.993	0.993	363703.000

Der Macro Average gibt die durchschnittlichen Werte von Precision, Recall und F1-Score über alle Klassen gleichgewichtet an, unabhängig von der Häufigkeit ihres Auftretens. Mit einem Makro-F1-Score von 0.961 zeigt das Modell, dass es auch weniger häufig vorkommende Labels gut erkannt hat. Der Weighted Average bezieht zusätzlich die Verteilung der Klassen im Datensatz mit ein. Hier wurde ein sehr hoher F1-Score von 0.993 erreicht, was darauf hinweist, dass häufiger vorhandene Entitäten tendenziell besser erkannt werden. Insgesamt zeigt die Kombination beider Metriken, dass das Modell sowohl häufige als auch seltene Entitäten zuverlässig klassifizieren kann.

Die nachfolgende Tabelle zeigt die Klassifikationsleistung des feinjustierten Modells auf den ursprünglich im Piiranha-Modell enthaltenen Entitäten. Die meisten Labels erreichen F1-Scores von über 0.90.

Abbildung 34: Bestehende Labels nach Fine Tuning des Modells

	precision	recall	f1-score	support
I-ACCOUNTNUM	0.878	0.903	0.890	2277.0
I-BUILDINGNUM	0.928	0.923	0.926	559.0
I-CITY	0.970	0.979	0.974	3672.0
I-CREDITCARDNUMBER	0.964	0.949	0.956	1291.0
I-DATEOFBIRTH	0.947	0.877	0.910	1613.0
I-DRIVERLICENSENUM	0.956	0.963	0.960	1197.0
I-EMAIL	0.996	0.989	0.992	7559.0
I-GIVENNAME	0.913	0.942	0.927	4174.0
I-IDCARDNUM	0.939	0.951	0.945	1499.0
I-PASSWORD	0.993	0.957	0.975	2010.0
I-SOCIALNUM	0.984	0.984	0.984	2255.0
I-STREET	0.986	0.963	0.974	1506.0
I-SURNAME	0.894	0.876	0.885	3120.0
I-TAXNUM	0.988	0.982	0.985	1972.0
I-TELEPHONENUM	0.993	0.998	0.996	4031.0
I-USERNAME	0.981	0.986	0.984	5478.0
I-ZIPCODE	0.954	0.967	0.960	1075.0

Besonders hohe Werte wurden bei den Klassen E-Mail und Telefonnummer erzielt, mit F1-Scores von jeweils über 0.99. Geringere Werte finden sich bei Accountnummer mit 0.890 und Nachname mit 0.885. Insgesamt befinden sich alle Labels in einem sehr guten Bereich, was zeigt, dass das Modell die bekannten Entitäten weiterhin zuverlässig klassifizieren kann.

Zur besseren Einordnung der Ergebnisse wurde das ursprüngliche Piiranha-Modell nochmals auf dem gleichen Testdatensatz evaluiert, der auch für das feinjustierte Modell verwendet wurde. So konnte die Klassifikationsleistung direkt vor und nach dem Fine Tuning verglichen werden. Die Labels REL, ETHN und SOR waren im ursprünglichen Modell nicht enthalten, weshalb deren F1-Score bei null ist.

Die folgende Tabelle zeigt den direkten Vergleich der F1-Scores beider Modelle. Dabei wird deutlich, dass sich die Leistung nicht nur bei den neuen Labels verbessert hat, sondern auch bei den ursprünglich enthaltenen Entitäten.

Tabelle 6: Performancevergleich der Modelle

Entität	F1-Score Piiranha-Modell	F1-Score feinjustiertes Modell
I-ACCOUNTNUM	0.788	0.890
I-BUILDINGNUM	0.741	0.926
I-CITY	0.852	0.974
I-CREDITCARDNUMBER	0.807	0.956
I-DATEOFBIRTH	0.699	0.910
I-DRIVERLICENSENUM	0.854	0.960
I-EMAIL	0.897	0.992
I-ETHN	0.000	0.957
I-GIVENNAME	0.753	0.927
I-IDCARDNUM	0.758	0.945
I-PASSWORD	0.868	0.975
I-REL	0.000	0.997
I-SOCIALNUM	0.906	0.984
I-SOR	0.000	1.000
I-STREET	0.834	0.974
I-SURNAME	0.692	0.885
I-TAXNUM	0.861	0.985
I-TELEPHONENUM	0.861	0.996
I-USERNAME	0.860	0.984
I-ZIPCODE	0.774	0.960
O	0.981	0.997

So stieg der F1-Score für Accountnummer von 0.788 auf 0.890 und auch der Erkennung des Nachnamens verbesserte sich von 0.692 auf 0.885. Besonders hohe Zuwächse sind zudem bei Geburtsdatum (+0.21) und Hausnummer (+0.18) zu beobachten.

Die Resultate der quantitativen Auswertung verdeutlichen, dass das Modell durch das Fine Tuning nicht nur erweitert, sondern auch in seiner Gesamtleistung besser wurde. Verschlechterungen konnten keine festgestellt werden. Die Erweiterung um neue Labels ging somit nicht zulasten der ursprünglichen Erkennungsfähigkeit.

Neben der stabilen Gesamtleistung zeigt sich insbesondere bei den neuen Entitäten ein klarer Mehrwert durch das Fine Tuning. Die drei neu eingeführten Entitäten Religion, Ethnie und sexuelle Orientierung wurden im ursprünglichen Modell nicht erkannt. Durch das gezielte Fine Tuning auf synthetisch erweiterten Trainingsdaten konnte das Modell diese Klassen erfolgreich erlernen.

Abbildung 35: Übersicht über die Performance auf neuen Labels

	precision	recall	f1-score	support
I-ETHN	0.969	0.946	0.957	166.0
I-REL	0.994	1.000	0.997	178.0
I-SOR	1.000	1.000	1.000	107.0

Die neu eingeführten Labels REL, ETHN und SOR wurden vom Modell mit sehr hoher Genauigkeit erkannt. Für REL liegt der F1-Score bei 0.997, für ETHN bei 0.957 und für SOR bei 1.000. Auch Precision und Recall bewegen sich bei allen drei Klassen auf einem sehr hohen Niveau.

Beim Label REL beträgt die Precision 0.994, der Recall liegt bei 1.000. Das bedeutet, dass alle relevanten Begriffe im Bereich Religion erkannt wurden und keine falschen Entitäten in diese Kategorie fielen. Für ETHN wurden mit einer Precision von 0.969 und einem Recall von 0.946 ebenfalls sehr gute Werte erreicht. Das Modell konnte hier die meisten Begriffe korrekt zuordnen, nur wenige relevante Entitäten blieben unerkannt. Besonders auffällig ist die Bewertung des Labels SOR, das mit einer Precision und einem Recall von jeweils 1.000 bewertet wurde. Das Modell hat demnach alle relevanten Begriffe zur sexuellen Orientierung korrekt erkannt und keine falschen Vorhersagen getroffen.

Diese Werte deuten auf eine sehr gute Modellleistung im Testdatensatz hin. Das Modell war in der Lage, die eingeführten Entitäten zuverlässig zu erkennen und klare Trennungen zwischen den verschiedenen Klassen vorzunehmen. Besonders im Hinblick auf die neu integrierten Kategorien zeigt sich, dass das Fine Tuning effektiv war.

Gleichzeitig ist zu berücksichtigen, dass es sich bei den Beispielen zu SOR, REL und ETHN ausschliesslich um synthetisch erzeugte Daten handelt. Die Testdaten folgten denselben Mustern und Strukturen wie die Trainingsdaten, was die hohen Ergebnisse teilweise erklären könnte. In realen Anwendungssituationen mit frei formulierten oder komplexeren Eingaben kann die Modellleistung jedoch abweichen.

Es ist anzunehmen, dass der Disentangled-Attention-Mechanismus von DeBERTa dazu beigetragen hat, eine zu starke Fixierung auf die Positionen der Entitäten zu vermeiden. Dennoch kann ein Overfitting einzelner Entitäten nicht ausgeschlossen werden, da die Anzahl der Trainingssätze begrenzt war und sich die Satzstrukturen stark ähnelten. Besonders bei SOR deuten die perfekten Werte darauf hin, dass sich das Modell an die bekannten Strukturen angepasst hat und bei neuen, natürlich formulierten Beispielen an seine Grenzen stossen könnte. Ein solches Risiko besteht bei REL in ähnlicher Form, fällt bei ETHN jedoch etwas geringer aus.

Nach der quantitativen Bewertung der Modellleistung folgt nun eine qualitative Betrachtung anhand konkreter Beispielsätze. Diese sollen veranschaulichen, wie sich die Erweiterung des Modells auf die tatsächliche Erkennung neuer Entitäten auswirkt und in welchen Fällen sich die Klassifikationsgenauigkeit verbessert hat. Dabei werden jeweils die Klassifizierungsergebnisse des ursprünglichen Piiranha-Modells denen des feinjustierten Modells gegenübergestellt.

Tabelle 7: Beispielsatz Modell-Vergleich Religion

Input	Ali ist Muslim und lebt in Zürich.
Piiranha-Modell	Ali (GIVENNAME): 0.55 Zürich (CITY): 0.96
Feinjustiertes Modell	Ali (GIVENNAME): 0.87 Muslim (REL): 1.00 Zürich (CITY): 0.98

In diesem Satz erkennt das ursprüngliche Piiranha-Modell lediglich den Vornamen und den Ort korrekt. Die Entität „Muslim“ wird nicht als Religion klassifiziert, da das ursprüngliche Modell diese Kategorie noch nicht kannte. Nach dem Fine Tuning erkennt das Modell die Entität „Muslim“ korrekt als REL mit hoher Wahrscheinlichkeit. Gleichzeitig bleiben die zuvor richtig identifizierten Entitäten nicht nur erhalten, sondern zeigen sogar eine höhere Confidence. Dies deutet darauf hin, dass die Erweiterung um neue Labels nicht auf Kosten der bestehenden Entitäten erfolgte.

Tabelle 8: Beispielsatz Modell-Vergleich sexuelle Orientierung

Input	Markus lebt offen als homosexuell und engagiert sich in der LGBTQ+-Community seiner Stadt.
Piiranha-Modell	Keine Entität erkannt.
Feinjustiertes Modell	Markus (GIVENNAME): 0.99 homosexuell (SOR): 1.00

Der Vorname Markus wurde vom ursprünglichen Modell nicht erkannt, obwohl er zur bestehenden Kategorie GIVENNAME gehört. Auch die Formulierung homosexuell wurde nicht identifiziert, da das ursprüngliche Labelset die Entität SOR nicht umfasste. Nach dem Fine Tuning gelingt es dem Modell, beide Entitäten zuverlässig zu identifizieren und korrekt zu klassifizieren. Das zeigt, dass das Modell nicht nur neue Labels wie SOR erkennen kann, sondern sich auch bei bestehenden Kategorien wie GIVENNAME verbessert hat.

Tabelle 9: Beispielsatz Modell-Vergleich Ethnie

Input	Elena Petrov, eine russische Teilnehmerin, hat die Emailadresse elena.petrov@uni-berlin.de angegeben.
Piiranha-Modell	Keine Entität erkannt.
Feinjustiertes Modell	Elena (GIVENNAME): 0.99 Petrov, (SURNAME): 1.00 russische (ETHN): 1.00 elena.petrov@uni-berlin.de (EMAIL): 0.99

In diesem Beispiel wird deutlich, dass das ursprüngliche Piiranha-Modell in einem komplex formulierten Satz keine der relevanten Entitäten erkennen konnte. Daraufhin wurde der Satz strukturell vereinfacht, um zu prüfen, ob das Modell unter einfacheren Bedingungen bessere Ergebnisse liefert.

Tabelle 10: Beispielsatz Modell-Vergleich Ethnie (reduzierte Komplexität)

Input	Elena Petrov hat die Emailadresse elena.petrov@uni-berlin.de angegeben.
Piiranha-Modell	Elena (GIVENNAME): 0.96 Petrov (SURNAME): 0.97 elena.petrov@uni-berlin.de (EMAIL): 0.99
Feinjustiertes Modell	Elena (GIVENNAME): 0.98 Petrov (SURNAME): 0.99 elena.petrov@uni-berlin.de (EMAIL): 0.78

Nach der Reduktion der Satzkomplexität konnte das Piiranha-Modell die Entitäten korrekt zuordnen. Das lässt vermuten, dass das Piirahna-Modell Schwierigkeiten mit komplexeren Satzstrukturen hat. Möglich ist auch, dass das komplexere Beispiel erst nach dem Fine Tuning erkannt wurde, weil solche Strukturen gezielt in den synthetischen Trainingsdaten enthalten waren. Es ist jedoch möglich, dass das feinjustierte Modell die strukturellen Schwächen des ursprünglichen Piiranha-Modells trotzdem teilweise übernommen hat. Die Herausforderungen bei komplexeren

Satzstrukturen könnten somit weiterhin bestehen, auch wenn durch das Fine Tuning zusätzliche Entitäten erkannt werden können.

Ein weiteres Ziel des Fine Tunings bestand darin, das Modell nicht nur auf konkrete Begriffe aus dem Trainingsdatensatz zu konditionieren, sondern auch eine gewisse Generalisierungsfähigkeit zu erreichen. Damit ist gemeint, dass das Modell in der Lage ist, auch abgewandelte oder zuvor nicht gesehene Entitäten korrekt zu erkennen. Wäre dies nicht gegeben, könnten auch einfache lexikon- oder regelbasierte Verfahren zum Einsatz kommen. Der Begriff Kosovarin war nicht Teil des Trainingsdatensatzes, wurde aber dennoch korrekt als ethnische Zugehörigkeit erkannt.

Tabelle 11: Generalisierungsfähigkeit des neuen Modells

Input	**Erkannte Entitäten**
Elena Petrov ist Kosovarin und hat die Emailadresse elena.petrov@uni-berlin.de angegeben.	Elena (GIVENNAME): 1.00 Petrov (SURNAME): 1.00 Kosovarin (ETHN): 1.00 elena.petrov@uni-berlin.de (EMAIL): 0.98

Das Modell zeigt, dass es nicht nur exakt bekannte Begriffe erkennt, sondern auch neue, inhaltlich ähnliche Ausdrücke korrekt zuordnen kann.

Die bisherigen Ergebnisse zeigen, dass das feinjustierte Modell viele Entitäten zuverlässig erkennen kann. Dennoch sind in bestimmten Kontexten auch Schwächen, insbesondere bei längeren, informationsdichten oder unstrukturierten Sätzen zu erkennen.

Tabelle 12: Klassifikation eines komplexeren Textabschnittes

Input	Erkannte Entitäten
Am 12. April 2024 meldete sich **Yasmin El-Haddad** telefonisch unter der Nummer **078 345 67 89** für das neue Diversity-Programm der Stadt Zürich an. In ihrer Anmeldung gab sie an, **muslimischen Glaubens** zu sein und sich aktiv in einer **LGBTQ+-Initiative** zu engagieren. Ihre Vorfahren stammen aus dem **nordafrikanischen Raum**, genauer gesagt aus **Marokko**. Sie wurde in der Schweiz geboren, besitzt aber zusätzlich einen deutschen Pass mit der Nummer **XJ1290035.** Ihre E-Mail-Adresse lautet **yasmin.elhaddad@posteo.de**. Bei der Anmeldung musste sie auch ihre Sozialversicherungsnummer **756.9234.5123.05** sowie ihre Adresse in der **Zähringerstrasse 29**, **8001 Zürich** angeben.	Yasmin (GIVENNAME): 1.00 El-Haddad (SURNAME): 1.00 078 345 67 89 (TELEPHONENUM): 1.00 **muslimischen Glaubens: False Negative** **LGBTQ+-Initiative: False Negative** **nordafrikanischen Raum: False Negative** **Marokko. (CITY): 0.68 False Positive (ETHN)** **XJ1290035: False Negative** **yasmin.elhaddad@posteo.de. (USERNAME): 0.99: False Positive (EMAIL)** 756.9234.5123.05 (SOCIALNUM): 1.00 Zähringerstrasse (STREET): 1.00 29, (BUILDINGNUM): 1.00 8001 (ZIPCODE): 0.99 **Zürich: False Negative**

Der Textausschnitt verdeutlicht die Grenzen des Modells bei längeren und komplexeren Eingaben. Während viele klassische Entitäten wie Name, Telefonnummer, Adresse oder Sozialversicherungsnummer zuverlässig erkannt wurden, zeigen sich hier bei inhaltlich dichten oder kulturell geprägten Passagen Schwächen. So wurden beispielsweise die Formulierung „muslimischen Glaubens“, „nordafrikanischen Raum“ und „LGBTQ+-Initiative“ nicht erkannt, obwohl sie einen Hinweis auf Religion, Ethnie und sexueller Orientierung geben. Zusätzlich wurden auch bekannte Kategorien wie CITY („Zürich“) und IDCARDNUM („XJ1290035“) nicht erkannt. Insgesamt blieben fünf Entitäten unerkannt und zwei wurden falsch klassifiziert. Die E-Mail-Adresse wurde fälschlich als USERNAME eingeordnet. „Marokko“ wurde als CITY gelabelt, obwohl es in diesem Kontext auf die Ethnie bezogen wäre.

Das Beispiel verdeutlicht, dass das Modell strukturierte Angaben grundsätzlich gut verarbeiten kann, bei umfangreicheren Eingaben mit komplexen oder impliziten Formulierungen jedoch fehleranfällig ist.

Der folgende Beispielsatz zeigt die Herausforderungen des Modells bei der Verarbeitung umgangssprachlich formulierter Texte.

Tabelle 13: Erkennung eines umgangssprachlichen Satzes

Input	**Erkannte Entitäten**
hab heute mit der neuen kollegin **malin** gesprochen, sie kommt ursprünglich aus dem **iran** und ist **muslimin**, deshalb will sie wegen **ramadan** flexible arbeitszeiten	malin (USERNAME): 1.00 **iran: False Negative** (ETHN): 0.49: **False Positive** muslimin, (REL): 0.86 **ramadan: False Negative**

In diesem Fall wurde „muslimin“ korrekt als REL klassifiziert, jedoch blieb die Herkunft „iran“ unberücksichtigt. Zwar wurde ein ETHN-Label vergeben, dies jedoch mit geringer Konfidenz und ohne klaren Bezug zu einem bestimmten Wort. Zudem wurde der Vorname „malin“ fälschlicherweise als USERNAME erkannt. Ebenfalls wurde das Wort „ramadan“ nicht als Religion klassifiziert beziehungsweise gar nicht als Entität erkannt. Diese Fehler deuten darauf hin, dass das Modell in weniger formellen Kontexten oder bei locker strukturierten Sätzen Schwierigkeiten hat, semantische Zusammenhänge korrekt zu erfassen.

Insgesamt zeigen die Ergebnisse, dass das feinjustierte Modell in wichtigen Punkten weiterentwickelt werden konnte. Die Einführung neuer Labels wie REL, ETHN und SOR war erfolgreich. Diese Entitäten wurden vom ursprünglichen Modell nicht erkannt, konnten nach dem Fine Tuning jedoch zuverlässig identifiziert werden. Besonders positiv ist, dass die Integration neuer Klassen nicht zulasten der bestehenden Erkennungsleistung ging. In mehreren Fällen konnten bestehende Entitäten im feinjustierten Modell sogar mit höherer Sicherheit erkannt werden.

Auch die quantitative Auswertung zeigt, dass das Modell auf den bereitgestellten Testdaten durchweg hohe Werte bei Precision, Recall und F1-Score erzielt. Dies unterstreicht die grundsätzliche Funktionsfähigkeit des erweiterten Modells. Gleichzeitig zeigen die qualitativen Beispiele, dass bei längeren, verschachtelten oder informellen Texten verschiedene Schwächen bestehen. Diese Schwächen wurden im Piiranha-Modell jedoch ebenfalls gefunden.

Das Modell liefert insgesamt überzeugende Resultate im Umgang mit klar strukturierten Daten. Gleichzeitig zeigen die qualitativen Beispiele, dass Schwächen bestehen. Diese Beobachtungen verdeutlichen, dass trotz der positiven Ergebnisse eine differenzierte Einordnung der Modellleistung notwendig ist.

5.4. Interpretation der Ergebnisse

Die Ergebnisse aus dem vorherigen Kapitel zeigen, dass das feinjustierte Modell eine insgesamt starke Klassifikationsleistung erzielt hat. Das gilt sowohl für die bereits bekannten Entitäten als auch für die neu eingeführten Kategorien Religion (REL), Ethnie (ETHN) und sexuelle Orientierung (SOR). Der hohe Macro Average und der Weighted Average verdeutlichen, dass sowohl häufige als auch seltenere Labels zuverlässig erkannt wurden.

Besonders positiv fällt auf, dass die neuen Kategorien der besonders schützenswerten Personendaten vom Modell grundsätzlich erkannt werden können. In eher klar strukturierten oder standardisierten Texten ist ein gezielter Einsatz bereits denkbar.

Gleichzeitig zeigt die qualitative Analyse, dass die Modellrobustheit bei realen oder komplexeren Texten begrenzt ist. In längeren oder umgangssprachlich formulierten Sätzen kam es häufiger zu Fehlklassifikationen oder Auslassungen. Auch vereinzelte Verwechslungen bei etablierten Entitäten, zeigen, dass das Modell besser auf sprachliche Vielfalt und natürliche Ausdrucksformen vorbereitet werden sollte. Diese Schwächen lassen sich auf die synthetischen Trainingsdaten zurückführen, da diese in ihrer Struktur und Ausdrucksweise stark standardisiert waren.

Die sehr guten Ergebnisse auf den synthetischen Testdaten deuten auf eine hohe Modellleistung hin. Da sich jedoch Trainings- und Testdaten stark ähneln, besteht das Risiko, dass das Modell eher die wiederkehrenden Strukturen der Daten gelernt hat als tatsächlich generell anwendbare Muster. In einem breiteren Sinne lässt sich dies als eine Form von Overfitting interpretieren, da die Leistung auf reale, variantenreiche Eingaben womöglich nicht übertragbar ist.

Die Ergebnisse zeigen, dass das Modell eine gute Grundlage für spezifische Anwendungsfälle bietet. Für den breiteren praktischen Einsatz ist jedoch eine höhere Robustheit und somit eine grössere Variation in den Trainingsdaten erforderlich. Eine gezielte Weiterentwicklung ist hier empfohlen.

5.5. Grenzen und Herausforderungen

Mehrere Rahmenbedingungen haben das Vorgehen bei der Modellentwicklung deutlich beeinflusst. Obwohl das Modell insgesamt gute Ergebnisse erzielen konnte, traten während des Entwicklungsprozesses verschiedene Herausforderungen auf, die sich auf die Leistungsfähigkeit ausgewirkt haben.

Die Rechenressourcen waren auf ein Endgerät beschränkt, weshalb ein vollständig neues Modelltraining nicht möglich war. Stattdessen musste auf ein vortrainiertes Modell zurückgegriffen werden, das seinerseits auf dem synthetischen AI4Privacy-Datensatz trainiert worden war.

Auch die Erweiterung des Modells um neue Labels unterlag Einschränkungen. Es standen keine öffentlich verfügbaren, gelabelten Datensätze mit besonders schützenswerten personenbezogenen Informationen zur Verfügung. Alternativ wurde geprüft, synthetische Daten mithilfe generativer KI zu erzeugen. Das KI-Modell setzte die Labels nicht immer korrekt. Eine manuelle Korrektur war im verfügbaren Zeitrahmen nicht machbar.

Deshalb wurde ein Template-basierter Ansatz zur Generierung der Trainingsdaten verwendet. Dieses Vorgehen ermöglichte präzise gesetzte Labels, führte jedoch zu einer begrenzten sprachlichen Variation, da die erzeugten Sätze häufig standardisiert

und formal blieben. Da auch das Ausgangsmodell auf synthetischen Daten trainiert wurde, wirkte sich die Kombination mit den künstlich erzeugten Beispielen der neuen Labels negativ auf die Modellleistung bei umgangssprachlichen oder komplexeren Sätzen aus.

Ein weiteres Hindernis war die sprachliche Beschränkung auf Deutsch. Aufgrund fehlender umfassender Kenntnisse in anderen Sprachen und Datenstrukturen anderer Regionen konnte das Modell nicht multilingual weiterentwickelt werden.

Die Evaluation des fertigen Modells erfolgte ausschliesslich auf einem synthetischen Testdatensatz. Dadurch kann keine genaue Aussage darüber getroffen werden, wie gut das Modell auf reale Daten übertragbar ist.

6. Diskussion und Ausblick

6.1. Beantwortung der Forschungsfrage

Die Ergebnisse dieser Arbeit zeigen, dass sich Named Entity Recognition gut eignet, um personenbezogene Daten in Texten automatisiert zu identifizieren. Dies gilt auch für besonders schützenswerte Kategorien, die im Rahmen dieser Arbeit erfolgreich in ein bestehendes Modell integriert wurden.

Um noch einmal explizit auf die Forschungsfrage zurückzukommen:

Wie kann Named Entity Recognition zur Klassifizierung personenbezogener Daten eingesetzt werden, um Unternehmen bei der DSGVO- und DSG-Konformität zu unterstützen?

NER eignet sich ausgezeichnet, um die Einhaltung datenschutzrechtlicher Vorgaben systematisch zu unterstützen. Dabei hat sich gezeigt, dass ein effizienter Ansatz darin besteht, vortrainierte Modelle wie Piiranha zu nutzen. Auf diese Weise lässt sich der anfängliche Trainingsaufwand reduzieren, während durch gezieltes Fine Tuning gleichzeitig die Erkennung spezifischer Entitäten erweitert und die Modellleistung verbessert werden kann.

Die Evaluation des feinjustierten Modells verdeutlicht, dass insbesondere standardisierte und klar strukturierte Texte zuverlässig klassifiziert werden können. Allerdings wurden auch Limitationen sichtbar, die in der praktischen Anwendung beachtet werden müssen. Vor allem bei längeren, komplexen oder umgangssprachlichen Texten steigt die Wahrscheinlichkeit von Fehlklassifikationen. Gerade im Kontext des Datenschutzes ist es entscheidend, dass automatisierte Verfahren eine möglichst geringe Fehlerrate aufweisen, da Fehler erhebliche Konsequenzen nach sich ziehen können, wie bereits in der Problemstellung dargelegt wurde.

Aus diesem Grund empfiehlt es sich, den Einsatz von NER-Modellen nicht isoliert, sondern stets im Zusammenspiel mit manuellen Kontrollmechanismen oder als

unterstützendes Instrument einzusetzen. Durch eine sorgfältige Evaluation des Modells auf realen Daten vor einem produktiven Einsatz lassen sich Fehlerrisiken reduzieren und die Modellleistung bei Bedarf vorgelagert verbessern.

Insgesamt bietet Named Entity Recognition eine praktikable Möglichkeit, Unternehmen effektiv bei der Umsetzung der Datenschutzkonformität zu unterstützen. Mit einer geeigneten Strategie zur Modellerweiterung und einem klaren Bewusstsein für die Grenzen der Technologie kann NER erheblich dazu beitragen, datenschutzrelevante Prozesse effizienter und sicherer zu gestalten.

6.2. Praktische Anwendung

Das trainierte Modell kann in verschiedenen Bereichen eingesetzt werden, in denen personenbezogene oder besonders schützenswerte Informationen verarbeitet werden. Der primäre Anwendungszweck liegt dabei in der automatisierten Klassifikation sensibler Daten, nicht in der direkten Umsetzung von Schutzmassnahmen. Dennoch kann das Modell Unternehmen gezielt dabei unterstützen, Datenschutzprozesse effizienter zu gestalten und regulatorische Anforderungen besser einzuhalten. Im Folgenden werden vier exemplarische Einsatzszenarien beschrieben, die den praktischen Nutzen des Modells veranschaulichen.

6.2.1. Schwärzung sensibler Inhalte

Ein denkbares Einsatzszenario ist die automatisierte Schwärzung sensibler Informationen in Textdokumenten. Hierzu könnten beispielsweise PDFs, E-Mails, Bilddateien oder andere unstrukturierte Texte bei Bedarf mittels OCR (Optical Character Recognition) in auswertbaren Text überführt und anschliessend vom Modell klassifiziert werden. Auf Basis der erkannten Entitäten liesse sich eine Schwärzung oder Markierung automatisiert umsetzen.

6.2.2. Klassifikation von Dokumenten

Das Modell kann zur Klassifikation der Vertraulichkeitsstufen von Dokumenten herangezogen werden. Ein Beispiel: Enthält ein Dokument ausschliesslich personenbezogene Daten, könnte es als „Vertraulich“ markiert werden. Werden

besonders schützenswerte Kategorien erkannt, wäre eine Einstufung als „Streng Vertraulich“ denkbar. Der Einsatz des Modells zur Klassifikation von Dokumenten hängt dabei stark von der Fehlertoleranz ab. In Szenarien mit hoher Toleranz, etwa bei der internen Klassifikation von E-Mails, können false negative Erkennungen akzeptabel sein. In Bereichen mit niedriger Fehlertoleranz, beispielsweise bei extern versendeten Daten, wäre hingegen eine zusätzliche manuelle Überprüfung erforderlich.

6.2.3. Systematische Übersicht über Datenbestände

Neben der Einzelverarbeitung eignet sich das Modell auch zur systematischen Analyse grösserer Datenmengen, beispielsweise in elektronischen Archiven oder Ablageordnern. In sensiblen Bereichen wie dem Finanz- oder Gesundheitswesen könnten regelmässige Scans bestehender Daten helfen, die Einhaltung von Datenschutzrichtlinien zu überprüfen. So liessen sich unbeabsichtigte Speicherungen sensibler Inhalte identifizieren oder potenzielle Verstösse frühzeitig erkennen.

6.2.4. Ergänzung eines bestehenden DLP-Systems

Das Modell kann auch in bestehende Data Loss Prevention- oder Compliance-Architekturen integriert werden. Es könnte dort als zusätzliche Erkennungsschicht dienen, beispielsweise in folgenden Kontexten:

- Data at Rest: Klassifikation gespeicherter Daten auf Dateiservern oder in Cloud-Umgebungen
- Data in Motion: Analyse von E-Mails oder Datenströmen während dem Versand
- Data in Use: Klassifikation sensibler Inhalte während der Bearbeitung von Dateien

Dabei ist wichtig zu betonen, dass das Modell kein vollständiges Data Loss Prevention System ersetzt. Es fungiert vielmehr als ergänzendes Klassifikationswerkzeug, das insbesondere da eingesetzt werden kann, wo regelbasierte Lösungen an ihre Grenzen stossen.

Bevor das Modell in einem der beschriebenen Anwendungsszenarien eingesetzt wird, sollte es auf realen Daten evaluiert werden. Nur so lässt sich beurteilen, wie gut die Klassifikation unter realen Bedingungen funktioniert und ob gegebenenfalls eine vorgängige Weiterentwicklung oder Anpassung notwendig ist.

6.3. Weiterentwicklung des Modells

Die bisherigen Ergebnisse zeigen, dass das feinjustierte Modell bereits eine solide Erkennungsleistung erreicht hat. Für den praktischen Einsatz in Unternehmen bestehen weitere Optimierungsmöglichkeiten. Diese betreffen zum einen die Weiterentwicklung des bestehenden Modells auf Basis der aktuellen Fähigkeiten, zum anderen die inhaltliche Erweiterung, um zusätzliche Entitäten zu erkennen.

Grosses Entwicklungspotenzial liegt in der weiteren Optimierung der Modellleistung bei den 20 aktuell bekannten Entitäten. Viele Begriffe wurden zwar zuverlässig erkannt, dennoch zeigten sich Schwächen bei umgangssprachlichen, komplexen oder unvollständig formulierten Sätzen. Um die Robustheit gegenüber sprachlicher Vielfalt zu erhöhen, empfiehlt sich eine Erweiterung der Trainingsdaten. Idealerweise werden diese mit realen, gelabelten Beispielen ergänzt. Sollte der Zugriff auf echte Daten nicht möglich sein, kann auch der bestehende Template-Ansatz weiter genutzt werden. In diesem Fall sollten die Satzmuster sprachlich vielfältiger ausgestaltet und die verwendeten Wortlisten inhaltlich erweitert werden.

Eine Möglichkeit zur kontinuierlichen Verbesserung wäre der Einsatz von Nutzerfeedback im Rahmen eines Human-in-the-loop-Ansatzes. Mitarbeitende könnten dabei automatische Klassifikationen manuell überprüfen und bei Bedarf korrigieren. Auf dieser Grundlage lässt sich das Modell schrittweise weiterentwickeln und in seiner Genauigkeit steigern.

Ein weiterer Entwicklungsbereich betrifft die sprachliche Abdeckung des Modells. Für international tätige Unternehmen ist es entscheidend, dass personenbezogene Daten in mehreren Sprachen erkannt werden können. Das ursprüngliche Piiranha-Modell ist bereits in der Lage, 17 Entitäten in sieben Sprachen zu identifizieren. Die Weiterentwicklung müsste sich daher auf die drei neu eingeführten Kategorien Religion, sexuelle Orientierung und Ethnie konzentrieren. Um dies zu ermöglichen, könnte der synthetische Datensatz gezielt um mehrsprachige Beispiele für diese drei Labels erweitert werden.

Auch die Ausrichtung auf unternehmensspezifische Inhalte bietet Potenzial. Mithilfe gelabelter unternehmensinterner Daten kann das Modell auf diese Inhalte angepasst werden. Dadurch lässt es sich besser auf die jeweilige Fachdomäne abstimmen und erkennt relevante Informationen noch zuverlässiger.

Die inhaltliche Erweiterung des Modells bietet weitere Chancen. So könnten zusätzliche besonders schützenswerte Kategorien aufgenommen werden, etwa Gesundheitsdaten, politische Meinungen oder die Zugehörigkeit zu einer Gewerkschaft. Dafür kann der Trainingsdatensatz mit entsprechenden Einträgen erweitert werden und das Modell, wie im Kapitel Fine Tuning beschrieben, erneut trainiert werden.

Parallel dazu könnte geprüft werden, ob bestimmte Entitäten nicht effizienter durch regelbasierte Ansätze erkannt werden können. Komplexe Kategorien wie Religion oder sexuelle Orientierung hängen stark vom sprachlichen Kontext ab. Strukturierte Informationen wie E-Mail-Adressen oder Sozialversicherungsnummern lassen sich dagegen zuverlässig mit regulären Ausdrücken erfassen. So könnten die Stärken von Transformer-Modellen gezielt für kontextabhängige Inhalte genutzt werden, während bei klar strukturierten Informationen der hohe Erkennungsgrad regelbasierter Methoden zum Einsatz kommt.

7. Fazit

Das zentrale Ziel dieser Arbeit bestand darin, ein bestehendes deutschsprachiges Named-Entity-Recognition-Modell um besonders schützenswerte Kategorien wie Religion, ethnische Herkunft und sexuelle Orientierung zu erweitern. Die quantitative Evaluation des feinjustierten Modells zeigte eine insgesamt hohe Leistungsfähigkeit und bestätigte die erfolgreiche Integration der neuen Kategorien. Praktische Tests verdeutlichten, dass das Modell in realen Kontexten Herausforderungen begegnet und hier an Genauigkeit verliert. Daher empfiehlt es sich, vor einem produktiven Einsatz eine ergänzende Evaluation mit realen Daten durchzuführen.

Es hat sich gezeigt, dass technische Lösungen wesentlich dazu beitragen können, rechtliche Anforderungen zuverlässiger einzuhalten. Die Entscheidung, synthetische Daten zu verwenden, schützte einerseits die Privatsphäre, verdeutlichte jedoch andererseits, dass qualitativ hochwertige künstliche Daten mehr sprachliche Vielfalt benötigen. Positiv konnte festgestellt werden, dass das Fine Tuning eines Modells auch mit begrenzter Hardware zu leistungsfähigen Modellen führen kann.

Insgesamt verdeutlicht diese Arbeit, dass ein gezielt erweitertes NER-Modell Unternehmen substanziell bei der effizienten Gestaltung datenschutzrelevanter Prozesse unterstützen kann. Die Ergebnisse bilden eine solide Grundlage für zukünftige Weiterentwicklungen. Damit steht ein praktikabler Ansatz zur Verfügung, um Klassifikationsmodelle gezielt im Kontext der Datenschutz-Compliance einzusetzen.

Abschliessend lässt sich festhalten, dass technologische Innovationen sinnvoll mit menschlicher Zusammenarbeit kombiniert werden können. Auch wenn Modelle wie das feinjustierte NER-Modell viele Aufgaben bereits automatisiert übernehmen können, bleibt insbesondere im Kontext von Datenschutz die sorgfältige Prüfung und Einordnung durch den Menschen unerlässlich.

8. Anhang

8.1. Abkürzungsverzeichnis

AI	Artificial Intelligence
BIO	Begin-Inside-Outside (Tagging)
CRISP-DM	Cross-Industry Standard Process for Data Mining
DLP	Data Loss Prevention
DSG	Datenschutzgesetz
DSGVO	Datenschutz-Grundverordnung
ETHN	Ethnie
EU	Europäische Union
F1	Harmonic Mean aus Precision und Recall
FN	False Negative
FP	False Positive
IO	Inside-Outside (Tagging)
KI	Künstliche Intelligenz
LLM	Large Language Model
ML	Machine Learning
NER	Named Entity Recognition
NLP	Natural Language Processing
OCR	Optical Character Recognition
PII	Personally Identifiable Information (personenbezogene Daten)
REL	Religion
SOR	Sexuelle Orientierung
TN	True Negative
TP	True Positive

8.2. Abbildungsverzeichnis

8.3. Tabellenverzeichnis

8.4. Quellenverzeichnis

Ai4privacy/pii-masking-400k · Datasets at Hugging Face. (2024, September 13). https://huggingface.co/datasets/ai4privacy/pii-masking-400k

Alammar, J., & Grootendorst, M. (2024). *Hands-on large language models: Language understanding and generation* (1st edition). O'Reilly.

Amaratunga, T. (2023). *Understanding Large Language Models: Learning Their Underlying Concepts and Technologies* (1st ed). Apress L. P.

Bahree, A., & Boyd, E. (2024). *Generative AI in action*. Manning Publications Co.

Barua, T., Hiran, K. K., Jain, R. K., & Doshi, R. (2024). *Machine Learning With Python* (1st Aufl.). De Gruyter.

Berwanger, D. D. J. (o. J.). *Definition: Datenschutz* [Text]. https://wirtschaftslexikon.gabler.de/definition/datenschutz-28043; Springer Fachmedien Wiesbaden GmbH. Abgerufen 3. November 2024, von https://wirtschaftslexikon.gabler.de/definition/datenschutz-28043/version-251682

Brown, B., & Zai, A. (2020). *Deep reinforcement learning in action*. Manning Publications Company.

Crowe, R., Hapke, H. M., Caveness, E., & Zhu, D. (2024). *Machine learning production systems: Engineering machine learning models and pipelines* (First edition). O'Reilly Media.

Delen, D. (2021). *Predictive analytics: Data mining, machine learning and data science for practitioners* (Second edition). Pearson.

Denham, E. (2018). *Information Commisssionier's Office; Democracy disrupted? Personal information and political influence*.

Eagar, G. (2023). *Data Engineering with AWS: Acquire the skills to design and build AWS-based data transformation pipelines like a pro* (1. Aufl.). Packt Publishing Limited.

Entwicklungsgeschichte der Datenschutz-Grundverordnung. (2018, Mai 25). https://www.edps.europa.eu/data-protection/data-protection/legislation/history-general-data-protection-regulation_de

Erik, H. (2025). *AI revealed: Theory, applications, ethics*. Mercury Learning and Information. https://doi.org/10.1515/9781501520679

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (Second edition). O'Reilly.

Géron, A. (2023). *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (Third edition). O'Reilly Media, Inc.

He, P., Liu, X., Gao, J., & Chen, W. (2021). *DeBERTa: Decoding-enhanced BERT with Disentangled Attention* (No. arXiv:2006.03654). arXiv. https://doi.org/10.48550/arXiv.2006.03654

Holzhofer, M. (2024). *DSGVO Bußgeld Datenbank—Immer aktuell und vollständig | dsgvo-portal.de*. https://www.dsgvo-portal.de/dsgvo-bussgeld-datenbank/

Huyen, C. (2025). *AI engineering: Building applications with foundation models* (First edition). O'Reilly.

Iiiorg/piiranha-v1-detect-personal-information · Hugging Face. (2025, Februar 23). https://huggingface.co/iiiorg/piiranha-v1-detect-personal-information

Kellerhals Carrard. (o. J.). Abgerufen 29. Oktober 2024, von https://kellerhals-carrard.ch/public/downloads/62bd91f280eb7.pdf

Khraisha, T. (2024). *FINANCIAL DATA ENGINEERING: Design and build data -driven financial products*. O'Reilly Media, Inc.

KMU Admin, K. M. U. (o. J.-a). *Datenschutz: Was man wissen muss*. Abgerufen 29. Oktober 2024, von https://www.kmu.admin.ch/kmu/de/home/aktuell/monatsthema/2023/datenschutz-was-man-wissen-muss.html

KMU Admin, K. M. U. (o. J.-b). *EU-Regelung zum Datenschutz: Neue EU-Verordnung*. Abgerufen 29. Oktober 2024, von https://www.kmu.admin.ch/kmu/de/home/fakten-und-trends/digitalisierung/datenschutz/datenschutz.html

Lammle, T., & Robb, D. (2024). *CCNA certification study guide volume 1: Exam 200-301* (Second). John Wiley and Sons.

LLC, C. T. (2025a). *Machine Learning Hero: Master Data Science with Python Essentials* (1st ed). Packt Publishing, Limited.

LLC, C. T. (2025b). *Natural Language Processing with Python: Master Text Processing, Language Modeling, and NLP Applications with Python's Powerful Tools* (1st ed). Packt Publishing, Limited.

Microsoft/mdeberta-v3-base · Hugging Face. (o. J.). Abgerufen 1. April 2025, von https://huggingface.co/microsoft/mdeberta-v3-base

OpenAI. (2024). *Chat GPT* (Version 4o) [Software]. OpenAI. https://chatgpt.com

Pai, S. (2025). *Designing Large Language Model Applications* (1. Aufl.).

Raschka, S. (2025). *Build a Large Language Model (from scratch)*. Manning.

Sarkar, D., Bali, R., & Sharma, T. (2018). *Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems* (1st ed. 2018). Apress : Imprint: Apress. https://doi.org/10.1007/978-1-4842-3207-1

Summary of the tokenizers. (o. J.). Abgerufen 11. Mai 2025, von https://huggingface.co/docs/transformers/tokenizer_summary

Tabrizi, R. (2025). *Behavioral AI: Unleash Decision Making with Data* (1st ed). John Wiley & Sons, Incorporated.

Taulli, T. (2025). *AWS CERTIFIED AI PRACTITIONER STUDY GUIDE: In-depth exam prep and practice*. O'REILLY MEDIA.

Thareja, R. (2024). *Artificial Intelligence: Beyond Classical AI*. https://learning.oreilly.com/library/view/artificial-intelligence-beyond/9789357053778/

Tierney, B. (2014). *Predictive analytics using Oracle data miner: Develop & use data mining models in Oracle Data Miner, SQL & PL/SQL*. McGraw-Hill Education.

Tunstall, L., Werra, L. von, & Wolf, T. (2022). *Natural language processing with transformers: Building language applications with Hugging Face* (First edition). O'Reilly Media.

Wie funktionieren Transformer-Modelle? - Hugging Face NLP Course. (o. J.). Abgerufen 1. April 2025, von https://huggingface.co/learn/nlp-course/de/chapter1/4

8.5. Beiliegende Dokumente

Die vollständige Implementierung dieser Arbeit befindet sich in mehreren Jupyter-Notebooks, die dieser Arbeit als digitale Beilage beigefügt sind.

Die einzelnen Notebooks haben folgende Funktionen:

- ai4privacy_datenbereinigung.ipynb – Bereinigung und Vorbereitung des AI4Privacy-Datensatzes
- ai4privacy_deskriptive_auswertung.ipynb – Grafiken des AI4Privacy Datensatzes
- syntdaten_erstellung.ipynb – Generierung synthetischer Trainingsdaten
- finetuning_modell.ipynb – Fine Tuning des NER-Modells

Zusätzlich sind folgende Datensätze enthalten:

- dataset_ai4privacy.csv – Bereinigter AI4Privacy-Datensatz
- dataset_synthetic.csv – Generierter synthetischer Datensatz

Das feinjustierte Modell ist auf der Plattform Hugging Face unter dem Namen „HuggingLil/pii-sensitive-ner-german“ veröffentlicht.