Projekt Group

Textmining in der Biomedizin

Autoren:

Guillermo García

Tim Ufer

Prüfer:

PD Dr. Martin Eisenacher

BetreuerInnen:

Anika Frericks-Zipper

Sai Spoorti Ramesh

Dr. Markus Stepath

ZUSAMMENFASSUNG

Biomarker sind messbare Indikatoren für klinische/medizinische Zustände wie Biomoleküle oder andere klinische Parameter (z. B. Blutdruck [1]), die zur Diagnose (diagnostisch [2]), Prognose (prognostisch [3]) oder Vorhersage (prädiktiv [3]) von Erkrankungen oder -im Fall der prädiktiven- auch von Therapieerfolgen genutzt werden. Proteine stehen im direkten Zusammenhang mit vielen Krankheiten (z. B. Brustkrebs [4]). Im Weiteren können für die Quantifizierung und Qualifizierung die Proteinmengen, beispielsweise mit Enzyme-Linked Immunosorbent Assay (ELISA) oder Massenspektrometrie (MS), gemessen werden. Proteine sind in ihren unterschiedlichen Funktionen oftmals in Krankheitsprozesse involviert, deshalb werden diese in wissenschaftlichen Fachzeitschriften als Biomarkerkandidaten veröffentlicht.

Allerdings ist die Information, welches Protein für welche Krankheit bereits veröffentlicht wurde, für alle Erkrankungen und Biomarker nicht immer frei und schnell zugänglich. Aus dieser großen Menge unstrukturierter Daten sollen die relevanten Informationen extrahiert und in aufbereiteter Form dem Nutzer in einer Knowledge Database zur Verfügung gestellt werden. Als potenzielle Lösung bietet sich die Biomarker Datenbank (DB) BIONDA an. In dieser Datenbank werden die relevanten Informationen aus frei zugänglichen Datenbanken für Publikationen aus den Bereichen Medizin und Biomedizin extrahiert und in eine SQL-Datenbank eingepflegt. Diese Datenbank erfasst alle publizierten Erkrankungen und die dazugehörigen potenziellen Protein-/Gen- oder mRNA-Biomarkerkandidaten. BIONDA beinhaltet desweiteren ein Scoring, also eine Möglichkeit, die die Qualität eines publizierten, vermeintlichen Kandidaten (statistisch) beurteilt.

Das Ziel dieser Arbeit ist die Erweiterung und Verbesserung der Funktionalitäten dieser Datenbank mit Techniken des machine-learnings (ML), in diesem Fall des Natural Language Processing (NLP). Mittels Named Entity Recognition (NER) sollen möglichst präzise Entitäten getaggt werden, um möglichst alle Biomarkerkandidaten zu finden. Dabei soll die falsch positiv Rate möglichst gering gehalten werden, um die Extraktion und Normalisierung von Proteinen/Genen und den dazugehörigen Krankheiten zu verbessern. Dafür werden die

Modelle in je zwei Datensätzen für Proteine/Gene und Krankheiten [5] wie z. B. NCBI-Disease getestet, und anschließend werden die Modelle untereinander und mit der BIONDA DB durch statistische Größen wie Precision, Recall und F1-Score verglichen.

ABSTRACT

Biomarkers are measurable indicators for clinical / medical conditions such as biomolecules or other clinical parameters (e.g. blood pressure [1]) that can be used for diagnosis (diagnostic [2]), prognosis (prognostic [3]) or prediction (predictive [3]) of diseases or - in the case of predictive ones - also for therapeutic successes. Proteins are directly related to many diseases (e.g. breast cancer [4]). Furthermore, the protein quantities can be measured for quantification and qualification, for example with Enzyme-Linked Immunosorbent Assay (ELISA) or mass spectrometry (MS). Proteins, in their various functions, are often involved in disease processes, which is why they are published in scientific journals as biomarker candidates.

However, information about which protein has already been published for which disease is not always accessible freely and quickly for all diseases and biomarkers. The relevant information is to be extracted from this large amount of unstructured data and made available to the user in a knowledge database in a preprocessed form. The biomarker database (BIONDA) is a potential solution. In this database, the relevant information is extracted from freely accessible databases for publications in the fields of medicine and biomedicine and entered into an SQL database. This database records all published diseases and the associated potential protein / gene or mRNA biomarker candidates. BIONDA also includes scoring, i.e. a possibility that (statistically) assesses the quality of a published, supposed candidate.

The aim of this thesis is to expand and improve the functionality of this database using machine learning (ML) techniques, in this case Natural Language Processing (NLP). Named Entity Recognition (NER) should be used to tag entities as precisely as possible in order to find as many biomarker candidates as possible. The false positive rate should be kept as low as possible to improve the extraction and normalization of proteins / genes and the associated diseases. For this purpose, the models are tested in two data sets for proteins / genes and diseases [5] such as NCBI disease. Then the models are compared with each other and with the BIONDADB using statistical values such as precision, recall and F1 score.

Inhaltsverzeichnis

Inhaltsverzeichnis **4**

1 EINLEITUNG **6**

1.1 Proteine, Gene und miRNA 6
1.2 Biomarker 7
1.3 Textmining 8
1.4 Biomedizinisches Textmining 9
1.5 BIONDA 11
1.6 Zielsetzung 11

2 MATERIALIEN UND METHODEN **12**

2.1 HunFlair 12
2.2 BioBERT 13
2.3 PubMedBERT 15
2.4 ScispaCy 16
2.5 de.NBI Cloud 17
2.6 Evaluation der einzelnen Modelle 17

3 IMPLEMENTATION DER TESTVERFAHREN **19**

3.1 Zusammenfügen der Datensätze 19
3.2 Training 20
3.3 Abruf der Abstracts über PubMed API 21
3.4 Performancetest in de.NBI Cloud 21
3.5 Berechnung von Precision, Recall und F1-Score 21

4 ERGEBNISSE **23**

4.1 Precision, Recall und F1-Score 23
4.2 Training und Testing auf biomedizinischen Datensätzen 24

4.3 Training und Testing auf Disease und Protein spezifischen Datensätzen . . . 25

5 DISKUSSION **27**

5.1 Software . . . 27
5.2 Modelle . . . 27
5.3 Precision vs. Recall . . . 28
5.4 BioBERT Mixed Domain vs. PubMedBERT Domain Specific . . . 28
5.5 HunFlair LSTM vs. PubMedBERT/ BioBERT Transformer . . . 30
5.6 CPU vs. GPU . . . 31
5.7 BIONDA NER vs. KNN NER . . . 31

6 AUSBLICK **33**

7 FAZIT **35**

Abkürzungsverzeichnis **36**

Literaturverzeichnis **38**

Abbildungsverzeichnis **42**

Tabellenverzeichnis **44**

Kapitel 1

EINLEITUNG

1.1 Proteine, Gene und miRNA

Proteine sind Makromoleküle welche aus, Aminosäuren bestehen, die durch Peptidbindungen verbunden sind. Mehrere dieser Aminosäuren, bilden ein Molekül, das als Polypeptid bezeichnet wird [6]. Proteine mit kompakter, kugelförmiger Struktur werden Globuläre Proteine genannt, wie z. B. Hämoglobin. Zu diesen Globulären Proteinen zählen unter anderem Enzyme und Antikörper [6]. Neben Globulären Proteinen existieren Strukturproteine. Diese eher linearen Proteine sind Bestandteile von z. B. Haaren und Muskelfasern [6]. Abhängig von der Art des Proteins reicht die Mutation einer Aminosäure aus um die Funktion oder die Struktur zu ändern [6]. Beispielsweise hat Vernon Ingram 1957 gezeigt, dass bereits die Veränderung einer Aminosäure des Proteins Hämoglobin verantwortlich für die Sichelzellkrankheit ist. Er konnte beweisen, dass die Mutation einer einzelnen Aminosäure ausreicht um einen pathologischen Effekt hervorzurufen [7].
Gene enthalten Informationen in Form von Desoxyribonukleinsäure (DNA). Damit genetische Informationen in molekulare Funktionen umgewandelt werden braucht es Proteine, ribosomale Ribonukleinsäuren (RNA) und Transfer-RNAs. Gene enthalten unter anderem die Informationen zum Bau von Proteinen [8]. Um messenger RNA (mRNA) zu synthetisieren, bindet die RNA-Polymerase an dem Promotor. Dadurch löst sich der DNA-Doppelstrang und die RNA-Polymerase liest den codogenen Strang in 3'-5' Richtung ab. An die DNA-Basen werden dann die komplementären RNA-Basen angelagert bis die RNA-Polymerase einen Terminator erreicht. Dieser Vorgang heißt DNA-Transkription.
Diese mRNA wird von einem Ribosom in 5'-3' Richtung gelesen. Das Ribosom besteht aus einer großen und einer kleinen Untereinheit. Die große Untereinheit hat eine A-Stelle, eine P-Stelle und eine E-Stelle. Eine Stelle wird jeweils von drei Basen der mRNA besetzt, die-

ses Triplet steht für eine Aminosäure. An der A-Stelle binden transfer RNAs (tRNA) mit der zum Triplet passenden Aminosäure, an der P-Stelle sitzt eine tRNA, mit welcher aus den angelieferten Aminosäuren eine Polypeptidkette gebildet wird. An der E-Stelle verlassen die tRNAs das Ribosom wieder. Dieser Vorgang wiederholt sich bis ein Stop-Codon auf der mRNA erreicht wird, dann zerfällt das Ribosom in seine Untereinheiten und entlässt die Polypeptidkette.
DNA besteht aus vier verschiedenen Nukleotiden Adenin, Thymin, Guanin und Cytosin. Das menschliche Genom besteht aus ca. 3 Milliarden Nukleotiden. Fast alle Erkrankungen des Menschen hängen mit Veränderungen in der Struktur und Funktion der DNA zusammen [9]. MicroRNAs (miRNA) sind eine Gruppe nichtkodierender RNAs mit einer Durchschnittslänge von 22 Nukleotiden, die eine wichtige Rolle bei der Genregulation spielen. miRNAs regulieren die Funktionsweise und Expression von mRNA. Die abnormale Expression von z. B. miRNA ist mit vielen menschlichen Krankheiten wie bspw. Lungenkrebs verbunden [10]. Die Erforschung von miRNA ist in den letzten Jahren in den Fokus der Wissenschaft gerückt; es wird vermutet, dass miRNA 20-30 Prozent der menschlichen Gene mitreguliert [11].

1.2 Biomarker

Die World Health Organisation (WHO) definiert Biomarker als "[...]any substance, structure or process that can be measured in the body or its products and influence or predict the incidence of outcome or disease."[1]. Biomarker sind nach der oben genannten Definiton messbare Indikatoren wie Biomoleküle oder andere klinische Parameter (z. B. Blutdruck oder Körpertemperatur), welche zu Diagnose, Verlauf oder Therapieerfolg von Krankheiten genutzt werden können. Proteine können aufgrund ihrer unterschiedlichen Funktionen im Organismus diagnostische (bspw. BRAC2 und Brustkrebs [4]), prognostische (bspw. TP53 und Chronisch lymphatische Leukämie [12]) oder predektive (TPMT und eine Behandlung mit Azathioprin [13]) Biomarker sein. Prognostische Biomarker werden verwendet um die Wahrscheinlichkeit eines klinischen Ereignisses vorauszusagen, z. B. das Wiederkehren oder das Fortschreiten einer Krankheit. Ähnlich dazu werden predektive Biomarker eingesetzt um die Wahrscheinlichkeit von positiven oder negativen Effekten, z. B. einer medizinischen Behandlung, für einen Patienten vorauszusagen. Dabei kann das Vorhandensein bzw. das Fehlen eines Markers Aufschluss über einen möglichen Therapieerfolg geben. Auch das Fehlen beispielsweise eines Proteins kann ein Biomarker sein [3]. Ein Biomarker kann sowohl prognostisch als auch predektiv sein (bspw. miR-21 und Mundhöhlenkarzinom [14]). Im Vergleich zu prognostischen und predektiven werden diagnostische Biomarker zur Erkennung

oder Bestätigung von Krankheiten eingesetzt. Hämoglobin A1c (HbA1c) ist ein molekularer diagnostischer Biomarker, welcher zusammen mit dem C-Peptid von Proinsulin eingesetzt werden kann um Patienten mit Typ-2 Diabetes mellitus (DM) zu identifizieren [2].

1.3 Textmining

Ein Ziel des Textminings (TM) ist es implizites Wissen aus unstrukturierten Texten abzuleiten und in expliziter Form darzustellen. Innerhalb des Textminings existieren viele verschiedene Ansätze und Vorgehensweisen. Eine davon ist *Named Entity Recognition*, mit der spezifische Begriffe in einem Text identifiziert und gefunden werden können [15]. NER-Techniken lassen sich in drei verschiedene Ansätze unterteilen: *dictionary-based, rule-based* und *machine learning*. Beim *dictionary-based approach* werden die gesuchten Begriffe anhand einer vorgegebenen Wortliste, z. B. Proteinlisten aus öffentlichen biologischen Datenbanken, identifiziert. Der *rule-based approach* verwendet vordefinierte Regeln zum Erkennen der Begriffe aus den Texten. Jedoch kann es auch hier passieren, dass die definierten Regeln nicht in allen Fällen effektiv sind. Zum Schluss gibt es noch den *machine-learning approach*. Bei diesem wird ein künstliches neuronales Netz (KNN) auf sog. Goldstandarddatensätzen trainiert, die von Menschen manuell kuriert worden sind. Im folgenden Teil des Kapitels wird der in dieser Arbeit genutzte *machine-learning approach* näher definiert und anhand von Beispielen erklärt. Neben dem Erkennen der Begrifflichkeiten müssen diese, um Synonyme zu erkennen und Dopplungen zu vermeiden, zusammengefasst werden. Dies ist Aufgabe der *Namend Entity Normalization* (NEN): beispielsweise sind '*epilepsy*' und '*falling sickness*' im Englischen zwei komplett unterschiedliche Wörter, die aber genau die gleiche Krankheit (Epilepsie) beschreiben [15]. Die oben beschriebe Problematik wird als *Entity Ambiguity* bezeichnet, hierbei können verschiedene Begriffe die gleiche Entität beschreiben oder ein Begriff kann, je nach Kontext, auf verschiedene Entitäten verweisen.

Eine weitere Aufgabe des Texminings ist die *Relation Extraction* (RE). Diese sucht nach Beziehungen zwischen verschiedenen Begriffen (z. B. Krankheiten und potenziellen Biomarker). Eine Möglichkeit dies umzusetzen ist die *Co-Occurrence* zu berechnen. Die Häufigkeit der *Co-Occurrence* kann dementsprechend einen Ansatz bieten, der die Beziehung zwischen Entitäten bewerten kann. *Syntactic parsing* ist ein weiterer Ansatz der RE. Hierbei wird die syntaktische Struktur des Satzes analysiert, um Abhängigkeiten und Beziehungen zwischen den Entitäten zu finden [16].

1.4 Biomedizinisches Textmining

Diese Arbeit konzentriert sich auf das Textmining von biomedizinschen Biomarkern aus der PubMed DB, da diese auch von BIONDA (Kapitel 1.5) genutzt wird, und die NER- sowie NEN-Methoden, welche im folgenden beschrieben werden, in BIONDA implentiert werden. Pubmed liefert den Großteil der für das BIONDA Projekt relevanten Publikationen aus Medizin und Biomedizin, welche frei und online zugänglich sind. Sie enthält zu den Publikationen Abstracts, Metainformationen und ein Tagging mit *Medical Subject Headings* (MeSH) Terms.

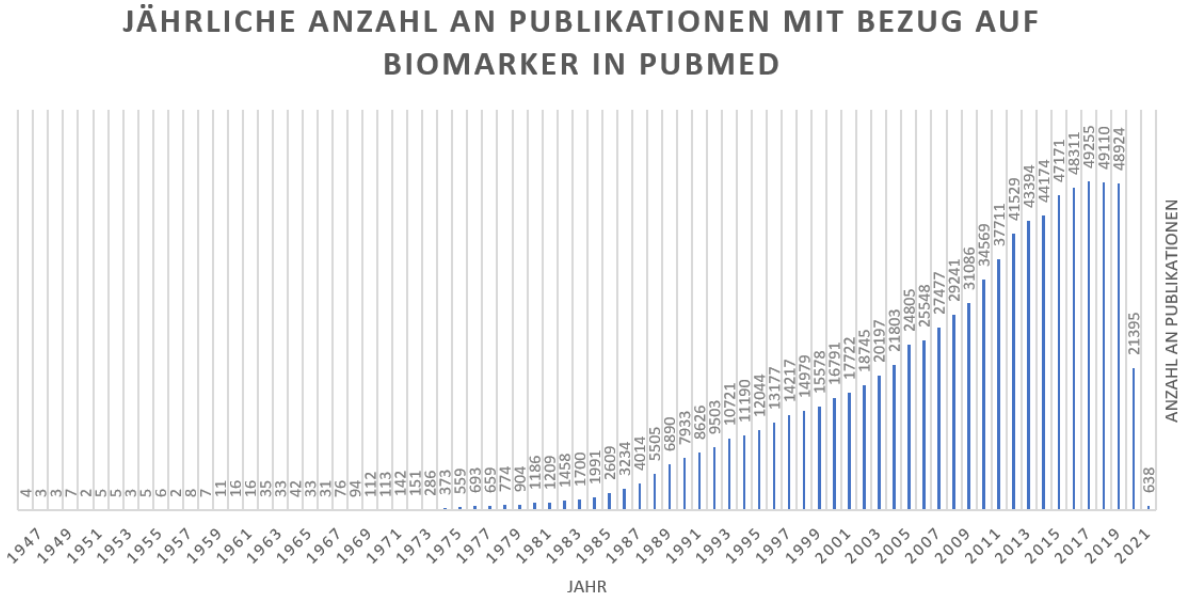

Abbildung 1.1: **Anzahl der Ergebnisse für die Suchanfrage "biological marker[MeSH Terms] & Human[organism]" in PubMed, für die Jahre 1946 bis 2021.** Die Y-Achse zeigt die jährliche Anzahl von Veröffentlichungen und die X-Achse die Jahre 1946 bis 2021. Bis 1985 steigt der Graph nur langsam an, danach steigt er deutlich stärker. Die Anzahl an Veröffentlichungen erreicht ihren Höhepunkt im Jahr 2019 mit 49255 [17].

Wie in Abbildung 1.1 zu erkennen, ist die Anzahl der veröffentlichen Publikationen in der Biomedizin zum Thema Biomarker seit 1985 deutlich gewachsen. Aufgrund der stark gestiegenen Anzahl an Publikationen ist es für Wissenschaftler nicht mehr möglich alle Publikationen zu lesen, auszuwerten und eventuelle Beziehungen zwischen den Ergebnissen verschiedener Publikationen zu finden. Ergebnisse zwischen den Publikationen zu vergleichen und Erkenntnisse daraus abzuleiten ist jedoch Grundlage medizinischen Fortschrittes. Als Konsequenz daraus ist die automatisierte Textanalyse nun ein in den letzten Jahren wichtig gewordener Parameter der biomedizinischen Forschung [16].

Das biomedizinische TM mit KNN kann in vier Phasen ablaufen, die anhand von Abbildung 1.2 näher erklärt werden.

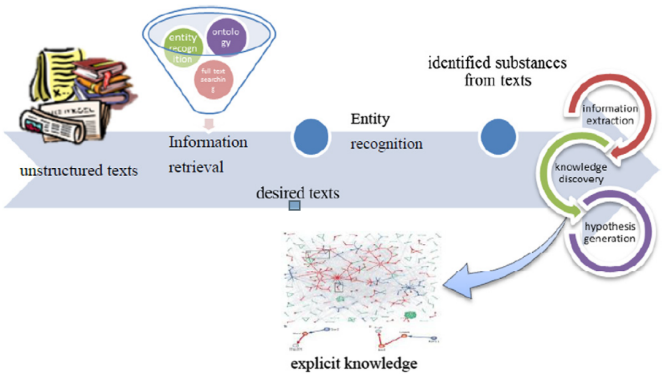

Abbildung 1.2: **Phasen und Aufgaben des Biomedizinischen Textminings mit KNN [15]**

In der ersten Phase, dem Informationenabruf (*Information retrieval*), werden relevante Texte anhand von Kriterien wie z. B. Themengebiet, Krankheiten, Genen etc. aus Datenbanken wie PubMed abgerufen. Anschließend werden die benötigten Informationen in Phase 2 aus den abgerufenen Texten extrahiert (*Information Extraction*). In dieser Phase werden mittels NER Entitäten aus dem Text markiert [15]. In der Fachliteratur wurden bereits verschiedene NER-Modelle speziell für biomedizinische Anwendungen publiziert, z. B. BioBERT [18], PubMedBERT [19], Stanza [20] oder ScispaCy [21]. Die dritte Phase ist die Wissensentdeckung (*Knowledge discovery*). Das Ziel dieser ist es Antworten auf biomedizinische Fragen anhand der Daten aus Phase eins und zwei zu finden, z. B. Ziele für neue Medikamente oder neue Biomarker zur Krankheitsdiagnose. In der vierten und letzten Phase, der Hypothesengenerierung (*Hypothesis generation*), werden Informationen, die nicht hinreichend erklärt sind oder weiterer Forschung bedürfen, als Hypothesen aufgestellt [15].

Für den Anwendungsfall der Nutzung eines KNN für biomedizinisches TM werden Datensätze für das Trainieren der Modelle benötigt. In der Literatur gibt es dazu ebenfalls bereits publizierte Datensätze wie NCBI-Disease [22] für Krankheiten oder JNLPBA [23] für Gene und Proteine. Wie bereits in Kap. 1.3 erläutet sind Namen oder Bezeichnungen in der Biomedizin nicht immer eindeutig, beispielsweise gibt es über 500 Synonyme für Ibuprofen [24]. Daher ist es nötig den Text mit Hilfe von NEN zu normalisieren. Zusätzlich zu NER und NEN ist RE ebenfalls ein wichtiger Bestandteil des biomedizinisches TM.

1.5 BIONDA

BIONDA[1] ist eine wissenschaftliche Datenbank, die Informationen über Krankheiten und dazugehörige potenzielle Biomarker kostenlos zur Verfügung stellt. Forschern, Kliniken und Mitarbeiter können diese Information nutzen. In der aktuellen Version von BIONDA werden die Biomarker und Krankheiten mit einem *dictionary-approach* erkannt, das heißt, dass die Entitäten als Folge von Buchstaben erkannt werden. Wenn sich ein Buchstabe änder, wird die Entität von der Methode nicht mehr erkannt. Dies verringert die Sensitivität beim Erkennen von Entitäten, da nur Entitäten erkannt werden welche, auch in dem *dictionary* enthalten sind.

1.6 Zielsetzung

Ziel der Arbeit ist es, mit Hilfe von *machine-learning* Methoden Verfahren zu testen, zu validieren und in BIONDA zu implementieren, die es erlauben, Biomarker mit höherer Präzision zu detektieren und die Detektion auch auf Biomarker auszudehnen, die bisher nicht entdeckt wurden. Um die Genauigkeit der Methoden zu messen, werden verschiedene Metriken verwendet, um effektiv zu vergleichen, ob die Anforderungen erfüllt werden. Dabei werden verschiedene Modelle für *machine-learning* verwendet, die im Folgenden beschrieben werden.

[1] http://bionda.mpc.ruhr-uni-bochum.de/start.php

Kapitel 2

MATERIALIEN UND METHODEN

In diesem Kapitel werden die vier Modelle des *machine-learning*-Ansatzes beschrieben, welche in dieser Arbeit verwendet werden und die Techniken aus Kapitel 1.3 verwenden. Weiterführend werden auch Evaluationsmethoden näher erläutert.

2.1 HunFlair

HunFlair [25] ist eine Erweiterung des Modells HUNNER [26], das ein halb-überwachtes *Pretraining* in *Pubmed* beinhaltet (Abbildung 2.1). Diese Modell basiert darauf, dass ein Satz versteckt wird, das Modell versucht, ihn vorherzusagen und dann die Parameter anhand der Fehler in der Vorhersage und des echten Satzes optimiert werden. Dafür wird das *Flair framework* [27, 28] verwendet. Wie HUNNER, ein eigenständiges NER-Tool mit *pre-trained* Modellen, verwendet es eine Kombination aus *bidirectional Long short-term memory* (LSTM), das nicht nur das aktuelle Wort berücksichtigt, sondern auch den Kontext, und *Conditional Random Fields* (CRF) zur Vorhersage von Entitäten. *Conditional Random Fields* sind eine Klasse statistischer Modellierungsmethoden, die sequentielle Abhängigkeiten in den Vorhersagen implementieren [29].

Das Training von HunFlair ist ein zweistufiger Prozess. Zuerst werden die erforderlichen *Word embeddings* auf einem großen unmarkierten Korpus trainiert, die dann im Training des NER-Taggers auf mehreren Goldstandarddatensätzen [1] verwendet werden. *Word embeddings* sind eine abstrakte Darstellung von Wörtern bzw. Strings als Zahlen in Vektorform.

Für die Flair-Einbettungen wird ein einschichtiges LSTM mit einer versteckten Größe von

[1] $https://github.com/flairNLP/flair/blob/master/resources/docs/HUNFLAIR_CORPORA.md$

2048 für jede Richtung verwendet. Die versteckte Größe ist definiert als die Layers zwischen der Eingabe und der Ausgabe. Es wird ohne explizite Vorstellung von Wörtern trainiert und modelliert daher grundsätzlich Wörter als Zeichenfolgen.

Für die FastText-Embeddings wird ein Skip-Gramm-Modell mit 200 Dimensionen trainiert. Skip-Gramm ist eine der unbeaufsichtigten Lerntechniken, mit denen die verwandtesten Wörter für ein bestimmtes Wort gefunden werden.

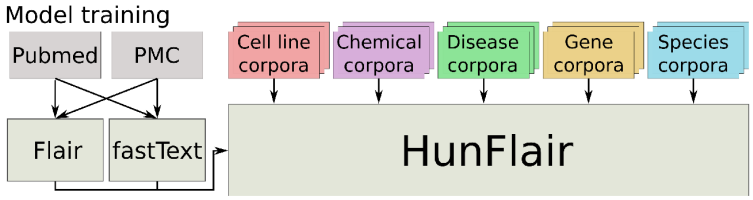

Abbildung 2.1: **HunFlair pretraining [25].** Es zeigt die Architektur von HunFlair. Der erste Teil basiert auf einem Sprachmodell auf Flair-Zeichenebene, das auf rund 24 Millionen Abstracts biomedizinischer Artikel von PubMed und 3 Millionen Volltexten aus PMC sowie auf FastText-Worteinbettungen trainiert wurde [30]. Der zweite Teil, die Vorhersage benannter Entitäten, wird von einem BiLSTMCRF-Modell durchgeführt. Analog zu HUNNER werden für jeden Entitätstyp unterschiedliche Modelle trainiert, wobei alle Trainingssätze aller Goldstandard-NER-Korpora mit diesem Typ kombiniert werden, um die Leistung gegenüber Textgenres und biomedizinischen Subdomänen zu verbessern.

2.2 BioBERT

BioBERT [18] ist ein vortrainiertes Sprachrepräsentationsmodell für den biomedizinischen Bereich. Zunächst wird BioBERT mit Gewichten aus dem BERT [31] initialisiert, das auf allgemeinen Domänenkorpora (englische Wikipedia Artikel und BooksCorpus) vortrainiert wurde. Anschließend wird BioBERT in biomedizinischen Domänenkorpora (PubMed-Abstracts und PubMed PMC-Volltextartikel) trainiert (Abbildung 2.2). Diese Art des Trainings wird *mixed-domain* Ansatz genannt, hierbei wird ein Modell erst auf allgemeinen und dann auf für den Aufgabenbereich spezifischen Texten trainiert.

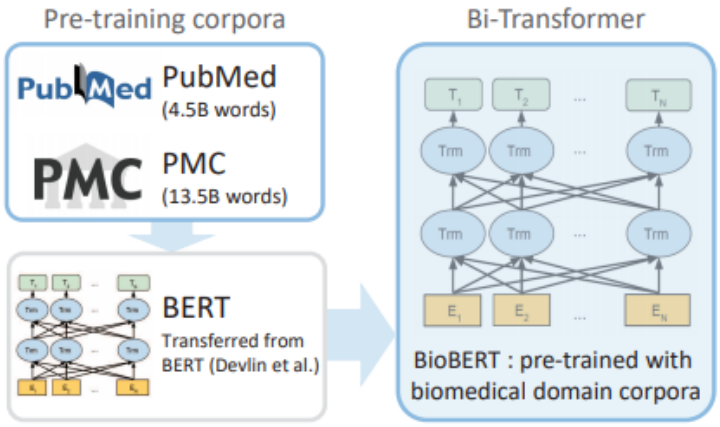

Abbildung 2.2: **Mixed-Domain Pretraining von BioBERT [18].** Zum Vortraining wird BioBERT mit den Parametern von BERT initialisiert, die in Wikipedia trainiert wurden. Von diesem Modell wird ein Pretraining mit Texten durchgeführt, die aus PubMed und PMC extrahiert wurden.

BERT, das Modell, auf dem BioBERT basiert, besteht aus einer *transformer* Architektur. Diese besitzt eine Serie von Encodern und Decodern. Die Encoder transformieren die Eingabe in eine interne Repräsentation, welche die Eingabe abstrakt abbildet, und die Decoder transformieren diese interne Repräsentation wieder in eine Ausgabe. Jedes Wort wird hierbei als gewichtete Summe der Werte der umliegenden Wörter dargestellt, so dass die gleichen Wörter je nach Kontext unterschiedliche Bedeutungen haben können. Diese Parameter werden vom Netzwerk während des Trainings durch *Gradient-Descent* erlernt [32]. *Gradient-Descent* ist ein iterativer Optimierungsalgorithmus zum Finden eines lokalen Minimums einer differenzierbaren Funktion. Die Idee ist, wiederholte Schritte in die entgegengesetzte Richtung des Gradienten (oder des ungefähren Gradienten) der Funktion am aktuellen Punkt zu machen, da dies die Richtung des steilsten Abstiegs ist. Es handelt sich um ein halbüberwachtes Training, welches mit der Technik des *Masked language modelling* durchgeführt wird. Hierbei werden einige Wörter oder Sätze versteckt und das Modell versucht, diese korrekt vorherzusagen. Danach kann das Modell *fine-tuned* werden, damit spezifische Aufgaben durchgeführt werden können. Beispielsweise können Aufgaben wie NER oder RE trainiert werden (Abbildung 2.3) [18].

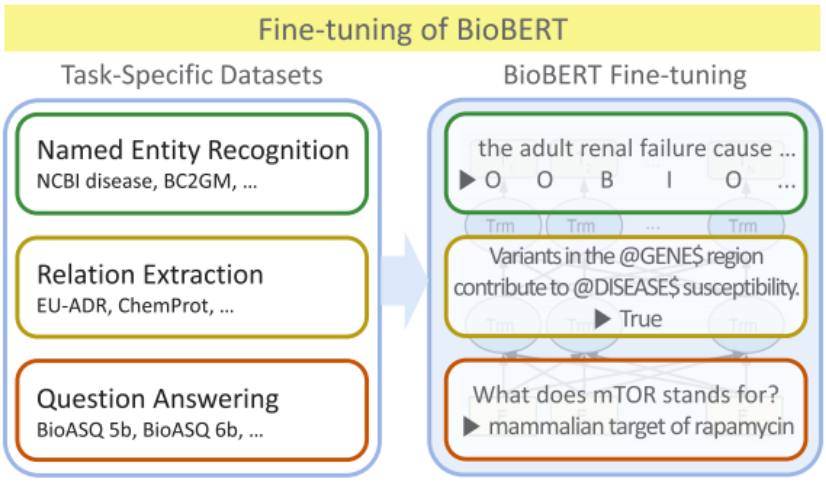

Abbildung 2.3: **Fine-tuning von BioBERT.** In BioBERT können unterschiedliche Aufgaben *fine-tuned* werden. In dieser Arbeit werden wir uns auf NER konzentrieren. [18]

2.3 PubMedBERT

PubMedBERT [19] ist ein domainspezifisch vortrainiertes Sprachrepräsentationsmodell auf Basis von BERT [31] für den biomedizinischen Bereich. Es unterscheidet sich von BioBERT (Kapitel 2.2) nur durch die Art des Vortrainings, das zugrunde liegende Modell BERT ist identisch. Im Gegensatz zu BioBERT, welches den in Abbildung 2.4 dargestellten *Mixed-Domain* Ansatz verwendet, bei welchem allgemeine und biomedizinische Domänkorpora genutzt werden, nutzt PubMedBERT *Domain-Specific pretraining* (Abb. 2.5).

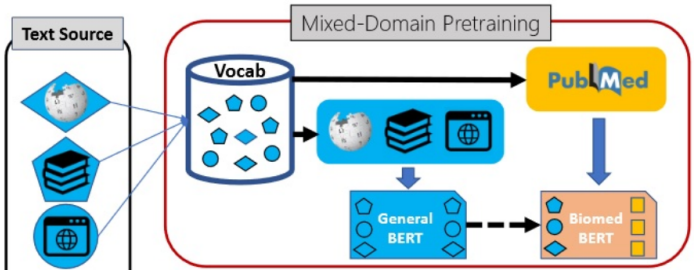

Abbildung 2.4: **Mixed-Domain Pretraining:** Das Pretraining wird erst auf einem Wortschatz aus z. B. Wikipedia und Büchern durchgeführt und anschließend auf Biomedizinischen Texten z. B. aus PubMed [19]

Bei dem *Domain-Specific* Ansatz aus Abbildung 2.5 wird das Modell nicht auf einer Mischung aus allgemeinen und biomedizinischen Texten traniert, sondern ausschließlich auf

biomedizinischen Texten. Hierzu wird eine Sammlung von 14 Mio. PubMed abstracts mit insgesamt 3.2 Mrd. Wörtern genutzt [19]. Die zugrunde liegende Annahme für das *Domain-Specific pretraining* ist, dass biomedizinische Texte sich stark von allgemeinen Texten unterscheiden und das Modell deshalb nicht durch ein *Mixed-Domain pretraining* profitiert. Das Training erfolgt dann, genauso wie in Kapitel 2.2 bereits erklärt, nur mit ausschließlich PubMed Abstracts.

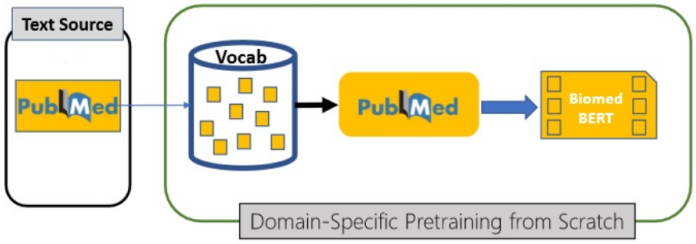

Abbildung 2.5: **PubMedBERT pretraining:** Das Pretraining wird auschließlich auf einem Wortschatz aus biomedizinschen Texten z. B. PubMed Abstracts durchgeführt [19]

2.4 ScispaCy

ScispaCy [21] ist ein Python Package, welches auf der *convolutional neural networks* (CNN) Architektur beruhende spaCy Modelle zur Verarbeitung biomedizinischer, wissenschaftlicher und klinischer Texte verwendet. Das NER Modell von ScispaCy wurde auf dem MedMentions [33] Datensatz trainiert, welcher aus 4.392 zufällig aus PubMed ausgewählten Artikeln (*title* und *abstracts*) besteht. Es nutzt ein transition-based System auf Grundlage des chunking Modells von Lample et al. [34]. Die Token werden innerhalb dieses Modells als gehashte und eingebettete Werte des Prefix, Suffix und der *lemmatized feature* der einzelnen Wörter dargestellt. Der Prefix sind Zeichen vor einem Wort wie (oder ", analog dazu sind Suffixe Zeichen am Ende eines Wortes wie km,), ", !. Die *lemmatized feature* ist die Grundform eines Wortes wie z. B. 'was' und 'be' oder 'rats' und 'rat'. Die NER Modelle in ScispaCy markieren die Entitäten, sagen jedoch, anders als Hunflair, BioBERT und PubMedBERT nicht deren Typ (Protein, Gene etc.) vorher. Für eine spezifische Erkennung (NER) einzelner Entitätsklassen (Protein, Krankheit, etc.) müssen jeweils individuelle hierfür trainierte Modelle separat verwendet werden. Diese wurden auf den gleichnamigen Goldstandarddatensätzen trainiert: BC5CDR (chemicals, diseases), CRAFT (cell types, chemicals, proteins, genes), JNLPBA (cell lines, cell types, DNAs, RNAs, proteins) and BioNLP13CG (cancer genetics) [21].

2.5 de.NBI Cloud

Das Deutsche Netzwerk für Bioinformatik-Infrastruktur (de.NBI[2]) ist ein nationales akademisches Netzwerk, das Serviceleistungen im Bereich Bioinformatik für die Lebenswissenschaften und Biomedizin in Deutschland und Europa anbietet. de.NBI bietet eine Cloud für das Hosting von *virtual machines* (VM) mit Ressourcen für Speicherplatz und Rechenleistung für die akademische Forschung an. Die VMs der de.NBI Cloud werden für diese Arbeit verwendet um den Ressourcenverbrauch (Zeit und Memory Auslastung) der im Kapitel 2 beschriebenen Modelle unter vergleichbaren Ressourcenvoraussetzungen zu evaluieren.

2.6 Evaluation der einzelnen Modelle

Um vergleichbare Ergebnisse für verschiedene Typen von Entitäten zu erhalten werden die Modelle auf den folgenden Datensätzen zur NER trainiert und getestet:

Datensatz	Typ
NCBI-disease	Disease
BC5CDR-disease:	Disease
BC5CDR-chem:	Drug/Chemical
BC4CHEMD:	Drug/Chemical
JNLPBA:	Gene/Protein
BC2GM:	Gene/Protein
LINNAEUS:	Species
S800:	Species

Tabelle 2.1: **Goldstandarddatensätze [5]**

Im Anschluss daran werden die Goldstandarddatensätze NCBI-disease, BC5CDR-disease, JNLPBA und LINNAEUS zu zwei großen Datensätzen für Diseases bzw. Proteine/Gene zusammengefasst, um die Modelle auf das Taggen von sowohl Genen/Proteinen als auch auf Krankheiten zu trainieren. Dazu wird ein in dieser Arbeit erstelltes Pythonskript genutzt, welches die TSV Dateien aller Datensätze in die einzelnen Sätze zerlegt. Im Anschluss daran werden die einzelnen Sätze in zufälliger Reihenfolge wieder zu einer TSV Datei zusammengesetzt. Zur Berechnung der Performance wird ebenfalls ein Pythonskript verwendet, welches die Prediction der Modelle mit dem Testteil der Datensätze vergleicht. Dieses Skript zählt die Anzahl der True Positives, True Negatives, False Positives und False Negatives. Aus diesen berechnet es den F1-Score, die Precision und den Recall. Zuletzt werden die Modelle auf

[2] www.denbi.de/

Raw-Data aus PubMed getestet um die Performance unter realen Bedingungen zu evaluieren und die Ergebnisse sowohl unter den Modellen als auch mit BIONDA zu vergleichen. Die Abtracts werden mithilfe von der PubMed Api automatisiert aus PubMed heruntergeladen.

Kapitel 3

IMPLEMENTATION DER TESTVERFAHREN

3.1 Zusammenfügen der Datensätze

Für das Zusammenfügen der Datensätze zum Training und Testing wurden zwei Python (3.8) Skripte geschrieben. Das erste Skript zerlegt die Datensätze in seine einzelnen Sätze und das zweite fügt die Sätze der beiden Datensätze wieder zusammen. Die Datensätze sind im TSV Format mit zwei Spalten 0 und 1 formatiert. In Spalte 0 steht je ein Wort und in Spalte 1 das entsprechende Tag, nach jedem Satz folgt eine Leerzeile. Die Tags haben das "BIO"Format (Beginning Inside and Outside of Entity) [35].

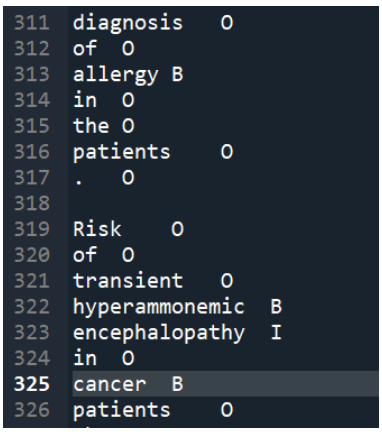

Abbildung 3.1: **Aufbau der TSV Datei.** Beispielhaft an dem NCBI-disease Datensatz

Das erste Skript durchläuft die TSV Zeile für Zeile und prüft ob es sich um eine Leerzeile handelt. Im falle einer Leerzeile wird die Nummer der Zeile gespeichert und alle Zeilen seit der letzten Leerzeile werden in eine neue TSV Datei kopiert und gespeichert. Im Anschluss

daran wird die aktuelle Leerzeile als die letzte Leerzeile gespeichert und das Programm wiederholt den Prozess bis das Ende der Datei erreicht wird. Das zweite Skript fügt die einzelnen Sätze der beiden Datensätze wieder in zufälliger Reihenfolge zusammen. Hierzu wird zunächst die Größe der beiden Datensätze zusammen errechnet und dann für jeden der beiden Datensätze der jeweilige Anteil an der Gesamtgröße. Dann wird durch den Pseudo-Zufallsgenerator *randrange(100)* eine zufällige Zahl zwischen 0 und 100 erzeugt. Anhand dieser Zahl und dem vorher errechneten Verhältnis der beiden Datensätze wird entweder ein Satz aus Datensatz 1 oder 2 an eine neue TSV Datei angehangen. Wenn zum Beispiel Datensatz 1 aus 60 Zeilen und Datensatz 2 aus 40 Zeilen besteht, ist das Verhältnis 60:40, d. h. wenn der Zufallsgenerator eine Zahl kleiner gleich 60 erzeugt wird ein Satz aus Datensatz 1 an die gemeinsame TSV Datei angehangen, anderenfalls einer aus Datensatz 2. Dies soll sicherstellen, dass die beiden Datensätze gleichmäßig auf den neuen Datensatz verteilt sind und so keinen Bias erzeugen. Das Programm wird solange ausgeführt bis alle Sätze aus Datensatz 1 und 2 in dem neuen Datensatz enthalten sind.

3.2 Training

Wie in Kapitel 2 erläutert, sind alle verwendeten Modelle auf großen Datenmengen (z. B. PubMedBERT auf 3,2 Mrd. Wörtern) vortrainiert. Zur Optimierung für finale Aufgaben wie NER werden die Modelle auf kleinen Mengen Trainingsdaten (120 Tsd. - 450 Tsd. Wörter) trainiert. Um die speziellen Aufgaben dieser Arbeit zu erfüllen, wird das Hunflair und BioBERT-Training wie folgt durchgeführt:

Zunächst werden die Trainingsdaten in .tsv-Dokumenten definiert, die wie erläutert ein Wort pro Zeile mit dem entsprechenden Label enthalten. Dann werden für jedes Wort die *WordEmbeddings* gebildet, die es erlauben, die Wörter in numerische Deskriptoren umzuwandeln. Die in jedem numerischen Deskriptor enthaltenen Informationen sind aufgrund des Vortrainings beschreibend für jedes Wort. Dann werden die Hyperparameter gemäß der Empfehlungen der Methode definiert. Das HunFlair-Modell wird für 200 Epochen mit einer Batchgröße von 32 und einer Lernrate von 0,1 trainiert. In Falle von PubMedBERT/BioBERT wird das Modell für 30 Epochen mit einer Batchgröße von 32 und einer Lernrate von 1e-5 trainiert. Dies entspricht den Empfehlungen für die jeweiligen Modelle [25][19][18].

3.3 Abruf der Abstracts über PubMed API

Zum Abrufen der Abstracts von PubMed wird die Python Library 'metapub' (0.5.5)[36] verwendet. Metapub ist eine Library zur Nutzung der PubMed API. Das Skript nutzt eine Liste von PubMed Abstract Nummern, die auch von BIONDA genutzt werden. Auf dieser Liste wird über die ersten 200 Abstracts iteriert und jeweils die Abstract Nummer an die Funktion *fetch.article_by_pmid()* übergeben. Diese gibt das Abstract zurück und der Text des Abstracts wird an eine Liste angehängt, die am Ende zurückgegeben wird und die Texte und Titel der ersten 200 Abstracts enthält.

3.4 Performancetest in de.NBI Cloud

Für die Messung des Ressourcenverbrauchs wird eine VM der de.NBI Cloud verwendet. Die gemessenen Ressourcen sind Zeit und Memory-Auslastung. Der Test wird auf 200 PubMed Abstracts durchgeführt, welche wie in Kapitel 3.3 beschrieben geladen werden. Diese werden dem Modell zur NER übergeben.

Zur Messung der Zeit wird das Python-Paket 'time'[1] verwendet. Dieses Paket bietet verschiedene zeitbezogene Funktionen. Mit dem Befehl *time.time*() wird die Startzeit des Modells festgehalten, die später von der Endzeit abgezogen wird um die Laufzeit zu erhalten.

Für die Messung der Memory-Auslastung wird die Python Library 'psutil'[2] (5.8.0) verwendet. Psutil ist eine Library zum Abrufen von Informationen zu laufenden Prozessen und zur Systemauslastung. Durch die Funktion *psutil.virtual_memory().percent* wird jeweils die prozentuale Memory-Auslastung für jeden Satz gemessen, der an das Modell übergeben wird.

3.5 Berechnung von Precision, Recall und F1-Score

Zur Berechnung der Metriken Precision, Recall und F1-Score wurde ein Skript geschrieben, welches die Prädiktion des Modells mit den Ergebnissen des Testteils des entsprechenden Datensatzes vergleicht und die Anzahl der True Positives, True Negatives, False Positives

[1] *https://docs.python.org/3/library/time.html*
[2] *https://github.com/giampaolo/psutil*

und False Negatives zählt.

Hierzu werden die TSV-Dateien der Prädiktion und des Datensatzes eingelesen und zeilenweise verglichen. Es werden für jedes Wort zeilenweise die entsprechenden BIO-Tags verglichen und für jedes Ergebnis wird die Variable des entsprechenden Wertes hochgezählt. Wenn alle Wörter verglichen sind werden Precision, Recall und F1-Score berechnet und zusammen mit vier anderen Werten ausgegeben. Die Formeln und Definitionen können den Tabellen 4.1 und 4.2 im folgenden Kapitel entnommen werden.

Kapitel 4

ERGEBNISSE

4.1 Precision, Recall und F1-Score

Um vergleichbare Ergebnisse zu erhalten wurden alle Modelle, wie in Kapitel 2.6 beschrieben, auf den Goldstandarddatensätzen trainiert und getestet. Als Bewertungsgrundlage werden die drei in Tabelle 4.1 aufgeführten Metriken Precision, Recall und F1-Score verwendet. Dabei gibt die Precision den Teil der richtig erkannten Elemente an, die relevant für die jeweilige Aufgabe sind, und der Recall den Teil der erkannten Elemente, die aus der Gesamtmenge aller für die Aufgabe relevanten Elemente erkannt wurden. Der F1-Score kombiniert Precision und Recall und gibt das gewichtete harmonische Mittel an. Auf diese Weise werden extreme bzw. weit auseinander liegende Werte "bestraft": wenn zum Beispiel eine der beiden Metriken sehr kleine Werte aufweist ist auch der F1-Score niedriger.

Die konkrete Berechnung der drei Metriken sowie die Definition der möglichen Ergebnisse True Positive, True Negative, False Positive und False Negative wird in den folgenden Tabellen dargestellt:

$$\text{Precision} = \frac{TruePositive}{TruePositive+FalsePositive}$$

$$\text{Recall} = \frac{TruePositive}{TruePositive+FalseNegative}$$

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

Tabelle 4.1: **Formeln für Precision, Recall und F1-Score**

	Vorhersage	Tatsächlicher Wert
True Positive	B	B
	I	I
	B	I
	I	B
True Negative	O	O
False Positive	B	O
	I	O
False Negative	O	B
	O	I

Tabelle 4.2: **Definition von True Positive, True Negative, False Positive und False Negative für das BIO Format.** Bei einem *Multiword Token* wird ein Treffer bereits als True Positiv gezählt, wenn nur ein Teil der Entität erkannt wurde.

4.2 Training und Testing auf biomedizinischen Datensätzen

Das folgende Kapitel beschreibt die Ergebnisse der in Kapitel 2 vorgestellten Methoden und Modelle. Um einen Überblick über die Performance der einzelnen Modelle zu erhalten wurden diese auf Datensätzen, aus dem biomedizinischen Bereich, für Diseases, Drugs/Chemicals, Genes/Proteins und Species trainiert (Kapitel 3.2) und getestet. Tabelle 4.3 zeigt die Ergebnisse jedes einzelnen Modells für alle Datasets der Typen mit den Metriken Precision, Recall und F1-Score. Die Ergebnisse für ScispaCy werden nicht betrachtet. Hunflair erreicht auf allen Datensätzen einen um 0,67-13,97 Prozentpunkte höheren F1-Score als BioBERT und PubMedBERT mit Ausnahme von den beiden Datensätzen für Drugs/Chemicals, wo PubMedBERT mit 89.32 % und 91.64 % den höchsten F1-Score erreicht. BioBERT erreicht nur auf den beiden Datensätzen für Species einen um 0,5 Prozentpunkte höheren F1-Score als PubMedBERT. Die Ergebnisse von BioBERT und PubMedBERT liegen für die meisten Datensätze nicht mehr als ± 2 Prozentpunkte auseinander, mit Ausnahme von BC2GM, wo der F1-Score von BioBERT um 2,65 Prozentpunkte niedriger ist als der von PubMedBERT. Der größte Unterschied in den Ergebnissen der drei Modelle findet sich bei den Datensätzen des Typs Genes/Proteins, wo Hunflair +11,92 (JNLPBA) bzw. +5,51 (BC2GM) Prozentpunkte über PubMedBERT liegt, und bei dem Datensatz S800, bei dem Hunflair +13,47 Prozentpunkte über BioBERT liegt.

Type	Dataset	Metric	BioBERT	HunFlair	PubMedBERT	ScispaCy
Diseases	NCBI-disease	Precision	86.60	**91.04**	87.13	-
		Recall	89.58	**93.77**	90.31	-
		F1-Score	88.07	**92.37**	88.69	-
	BC5CDR-disease	Precision	84.84	**88.99**	86.61	-
		Recall	87.95	87.78	**88.81**	-
		F1-Score	86.37	**88.37**	87.70	-
Drugs/Chemicals	BC5CDR-chem	Precision	92.66	91.77	**93.57**	-
		Recall	93.42	93.15	**95.00**	-
		F1-Score	93.04	92.45	**94.28**	-
	BC4CHEMD	Precision	91.74	89.65	**91.94**	-
		Recall	90.57	89.65	**91.25**	-
		F1-Score	91.15	89.65	**91.59**	-
Genes/Proteins	JNLPBA	Precision	70.36	**86.66**	71.86	-
		Recall	82.58	**92.38**	83.86	-
		F1-Score	75.98	**89.32**	77.40	-
	BC2GM	Precision	82.62	**91.85**	84.24	-
		Recall	84.22	**91.40**	85.89	-
		F1-Score	83.41	**91.63**	85.05	-
Species	LINNAEUS	Precision	90.93	**97.48**	88.96	-
		Recall	**86.11**	82.77	85.84	-
		F1-Score	88.45	**88.72**	87.18	-
	S800	Precision	70.91	**87.48**	69.25	-
		Recall	76.92	**87.05**	77.83	-
		F1-Score	73.79	**87.26**	73.29	-

Tabelle 4.3: **Macro-Ergebnisse der Modelle.** Ergebnisse der Modelle BioBERT, Hunflair und PubMedBERT mit den Parametern Precision, Recall und F1-Score, nach dem Training auf den unter Datasets angegebenen Goldstandarddatensätzen.

4.3 Training und Testing auf Disease und Protein spezifischen Datensätzen

Da sich diese Arbeit auf Krankheiten und Biomarker beschränkt, wurden die drei Modelle, wie in Kapitel 2.6 beschrieben, im Anschluss auf zwei zusammengesetzten Datensätzen für Proteine/Gene (JNLPBA und BC2GM) bzw. Diseases (NCBI-disease und BC5CDR-disease) trainiert und getestet.

Tabelle 4.4 gibt die Ergebnisse von BioBERT, HunFlair, PubMedBERT und BIONDA auf dem kombinierten Datensatz für Diseases wieder. BIONDA erreicht mit 98.66 % die höchste Precision, während HunFlair mit 83.29 % die niedrigste Precision erreicht. Für die

übrigen Metriken Recall und F1-Score erreicht jeweils PubMedBERT mit 90.04 % und 89.47 % die höchsten und BIONDA mit 15.99 % und 27.53 % die niedrigsten Werte. Während die Modelle des *machine-learning approach* den höchsten Unterschied mit ±5.62 Prozentpunkten bei der Precision aufweisen, ist das Recall-Ergebnis von BIONDA mit 15,99 % um 69,91 Prozentpunkte niedriger als das des niedrigsten Modells HunFlair mit 85.90 %. Auch der F1-Score liegt im Schnitt um 59.86 Prozentpunkte niedriger als der der Modelle des *machine-learning approach*. BioBERT erreicht von den drei *machine-learning approach* Modellen mit 88.06 %, 88.15 % und 88.10 % in allen Metriken die niedrigsten Werte.

Type		Metric	BioBERT	HunFlair	PubMedBERT	BIONDA
Results	Metrics	Precision	88.06	83.29	88.91	**98.66**
		Recall	88.15	85.90	**90.04**	15.99
		F1-Score	88.10	84.58	**89.47**	27.53

Tabelle 4.4: **Ergebnisse der Modelle für Precision, Recall und F1-Score nach dem Training auf den kombinierten Datensätzen für den Entitätentyp Diseases**

Tabelle 4.5 zeigt die Ergebnisse von BioBERT, HunFlair, PubMedBERT und BIONDA auf dem kombinierten Datensatz für Proteine/Gene. Auf diesem Datensatz erreicht HunFlair in allen Metriken mit 89.01 %, 91.76 % und 90.36 % die höchsten Werte. BIONDA erreicht im Vergleich die niedrigsten Werte mit 59.65 %, 4.97 % und 9.17 %. Von den drei *machine-learning approach* Modellen erreicht BioBERT auch auf diesem Datensatz wieder die niedrigsten Werte mit 82.57 %, 87.28 % und 84.85 %. BioBERT, HunFlair und PubMedBERT weisen den größten Unterschied mit ±6.44 Prozentpunkten bei der Precision auf. Der größte Unterschied zwischen BIONDA und dem niedrigsten Ergebnis der anderen drei Modelle liegt bei 82.31 Prozentpunkten bei dem Recall. Der F1-Score von BIONDA ist im Durchschnitt um 77.74 Prozentpunkte niedriger als bei den anderen drei Modellen.

Type		Metric	BioBERT	HunFlair	PubMedBERT	BIONDA
Results	Metrics	Precision	82,57	**89.01**	83,78	59.65
		Recall	87,28	**91.76**	87,98	4.97
		F1-Score	84,85	**90.36**	85,82	9.17

Tabelle 4.5: **Ergebnisse der Modelle für Precision, Recall und F1-Score nach dem Training auf den kombinierten Datensätzen für den Entitätentyp Proteine/Gene**

Kapitel 5

DISKUSSION

5.1 Software

Dieser Abschnitt der Diskussion beschäftigt sich mit der Frage welche Software zur Umsetzung dieses Projektes verwendet wurde.

Alle Skripte dieser Arbeit wurden in Python und mit Hilfe von Python Libraries geschrieben. Python unterstützt eine Vielzahl von *machine-learning* und *natural language Processing* libraries und toolkits. Außerdem sind die in dieser Arbeit verwendeten Modelle alle für Python ausgelegt [18] [19] [25]. Python ist weit verbreitet für die Nutzung von *machine learning* Modellen. Außerdem wird Python auch in der Arbeitsgruppe genutzt, was eine Weiterführung des Projektes vereinfacht.

Als Entwicklungsumgebung wurde Anaconda's Spyder[1] genutzt, da diese für die wissenschaftliche Programmierung entwickelt wurde und eine Vielzahl von Paketen, wie z. B. *pandas*, unterstützt.

5.2 Modelle

Im folgendem wird die Wahl der Modelle behandelt, die in dieser Arbeit verwendet werden. Aufgrund des begrenzten Zeitrahmens wurde nach einer Literaturrecherche eine Auswahl der für die Arbeit am besten passenden Modelle vorgenommen. Die Wahl fiel auf die vier Modelle BioBERT, PubMedBERT, Hunflair und ScispaCy. Nach der Literaturrecherche und einigen Tests wurde ScispaCy nicht weiter in Betracht gezogen, da ScispaCy zwar NER Aufgaben

[1] $https://www.spyder-ide.org/$

auf Texten ausführen kann, dabei aber nicht den Typ der markierten Entitäten bestimmt [21]. Dadurch war ScispaCy nicht für die Aufgaben geeignet, die in dieser Arbeit umgesetzt werden sollen, da es keine Unterscheidung von Proteinen, Krankheiten, Medikamenten etc. zulässt. Von den drei verbleibenden Modellen wurden die beiden BERT Modelle ausgewählt, da die Transformer Architektur auf der diese Modelle basieren, in den letzten Jahren mehrfach für ähnliche Zwecke publiziert wurde. Zudem existieren bereits vortrainierten Modelle für die Anwendungszwecke dieser Arbeit. Hunflair wurde ausgewählt, um die Transformer Architektur zusätzlich mit der *Long short-term memory* Architektur zu vergleichen und eventuelle Unterschiede in den Ergebnissen zu untersuchen.

5.3 Precision vs. Recall

Die in dieser Arbeit verwendeten Metriken Precision und Recall besitzen je nach Endbenutzer eine unterschiedliche Relevanz. Im biomedizinischen Kontext sind die Endbenutzer der BIONDA DB - und damit indirekt auch der in dieser Arbeit vorgestellten Modelle - Mediziner/Wissenschaftler. Diese sind mehr an der Qualität als an der Quantität der Treffer interessiert, da ein qualitativ hochwertiger Biomarker eine höhere Aussagekraft über Krankheiten besitzt, als eine Vielzahl qualitativ minderwertiger Marker. Daher hat in diesem Kontext die Precision eine höhere Relevanz als der Recall. Desweiteren werden auch mit einem niedrigen Recall viele Treffer erzielt, da eine sehr große Menge an Abstracts zur Verfügung steht.

5.4 BioBERT Mixed Domain vs. PubMedBERT Domain Specific

Dieser Abschnitt untersucht, ob *Domain-specific Pretraining* bei NER Aufgaben tatsächlich eine Verbesserung im Vergleich zu *Mixed-Domain Pretraining* darstellt, wie in Yu Gu et al [19] behauptet.

Im Vergleich mit der Publikation von Yu Gu et al [19] nähern sich die hier erzielten Ergebnisse den publizierten an und stellen zum Teil eine Verbesserung dar:

Dataset	PubMedBERT F1-Score Publikation [19]	Studienprojekt
BC5-chem	93.33	94.28
BC5-disease	85.62	87.70
NCBI-disease	87.82	88.69
BC2GM	84.52	86.06
JNLPBA	80.06	77.40

Tabelle 5.1: **F1-Score von PubMedBERT aus der Publikation [19] und dieser Arbeit für NER auf den Testteilen der verschiedenen Datensätze**

Anhand von Tabelle 5.1 lässt sich erkennen, dass das PubMedBERT Modell in dieser Arbeit auf allen Datensätzen, außer JNLPBA, sogar 0,95 bis 2,08 Prozentpunkte besser abschneidet als das Modell aus der Publikation [19]. Dies lässt sich jedoch durch verschiedene Trainingsparameter erklären. In der Publikation [19] werden eine *learning rate* von (1e-5, 3e-5, 5e-5), *batch size* (16, 32) und *epoch number* (2-60) benutzt. Dort wird allerdings angemerkt, dass diese Parameter nicht optimal sind und lediglich gewählt wurden um für alle Modelle der Publikation einheitliche Parameter zu nutzen.

In einem weiteren Vergleich werden die Ergebnisse des BioBERT Modells dieser Arbeit dem der Publikation [19] gegenübergestellt:

Dataset	BioBERT F1-Score Publikation [19]	Studienprojekt
BC5-chem	92.85	93.04
BC5-disease	84.70	86.37
NCBI-disease	89.13	88.07
BC2GM	83.82	83.41
JNLPBA	79.35	75.98

Tabelle 5.2: **F1-Score von BioBERT aus der Publikation [19] und dieser Arbeit für NER auf den Testteilen der verschiedenen Datensätze**

Auch die Ergebnisse aus Tabelle 5.2 weisen eine geringe Abweichung auf mit dem größten Unterschied bei dem Datensatz JNLPBA mit 3,37 Prozentpunkten. Auch wenn die Ergebnisse von BioBERT und PubMedBERT in dieser Arbeit, mit Ausnahme von JNLPBA, nur 1-2 Prozentpunkte auseinander liegen daraus lässt, sich daraus ableiten, dass PubMedBERT durch das *Domain-Specific pretraining* einen Vorteil gegenüber BioBERT hat. In der Publikation [19] wird ebenfalls erläutert, dass BioBERT davon profitiert, dass es auf biomedizinischen Texten trainiert wurde, aber dies das Training auf dem allgemeinen Domänkorpora

nicht vollständig ausgleicht und deshalb die Ergebnisse näher bei einander liegen [19].

5.5 HunFlair LSTM vs. PubMedBERT/ BioBERT Transformer

In dieser Arbeit wurden zwei der aktuellsten Architekturen im NLP-Bereich betrachtet: LSTM und Transformers. In der vorliegenden Arbeit wurden die besten in Kapitel 4 beschriebenen Ergebnisse mit der LSTM-Architektur von HunFlair erzielt, welche bessere Ergebnisse erbrachte als die Transformer Architektur von BioBERT und PubMedBERT. Dies kann darauf zurückzuführen sein, dass in der LSTM-Architektur im Vergleich zur Transformer Architektur jedes Wort viel stärker durch seine nächste Umgebung definiert wird. Die Merkmale jedes Wortes sind bei diesem Verfahren nach dem Durchlaufen des neuronalen Netzes die gewichtete Summe der vorherigen und nachfolgenden Wörter (Bi-LSTM), so dass die dem Wort am nächsten liegenden Wörter dieses mehr beeinflussen. Dies ist hier von großer Bedeutung, da die Biomarker- und Krankheitsbegriffe stark durch ihre nächste Umgebung charakterisiert sind (z. B. wird das Wort "Krebs" neben "Lunge" stehen, um zu spezifizieren, um welchen Begriff es sich handelt).

Im Gegensatz dazu wird bei der Transformer Architektur der gesamte Kontext (200 Token) eines Satzes betrachtet. So werden die Merkmale jedes Wortes bestimmt und dadurch erkannt, welche Wörter aus dem Kontext die notwendigen Informationen liefern, um sie als Biomarker oder Krankheiten zu klassifizieren. Aufgrund der Vielzahl an möglichen Kombinationen werden die Merkmale jedes Wortes nicht von der nahen Umgebung bestimmt, sondern von der Summe derjenigen, die im gesamten Satz als am wichtigsten angesehen werden. Diese Eigenschaft, die bei anderen Aufgabentypen, bei denen die Information an beliebiger Stelle im Satz stehen kann, sehr vorteilhaft ist, verringert die Genauigkeit dieser Aufgabe im biomedizinischen Bereich etwas [25]. Daher erreicht HunFlair einen im Schnitt um 4,15 Prozentpunkte höheren F1-Score (Tabelle 4.3) als die Transformer Modelle BioBERT und PubMedBERT. Allerdings liefern beide Architekturen F1-Scores über 90 % (Tabelle 4.3) und sind für diese Aufgabe geeignet.

5.6 CPU vs. GPU

In Unterkapitel 5.6 wird die Entscheidung zur Nutzung einer GPU anstelle einer CPU diskutiert. Die Modelle wurden zuerst auf den Grafikkarten (engl. *graphics processing unit*, GPU) von privaten Computern trainiert. Da dies zwischen vier und zwölf Stunden dauern konnte, abhängig von der Größe des Datensatzes und der Auslastung des Computers, wurde versucht die Modelle in der de.NBI Cloud mit 8 virtuellen Prozessoren (engl. *virtuell central processing units*, VCPU) zu trainieren, da die Computer nicht baugleich waren und daher keine gleichbleibenden Konditionen für das Training gegeben waren.

Dabei stellte sich heraus, dass das Training eines Datensatzes, das auf der GPU ca. 8 Std. gedauert hatte, mit 8 VCPUs über 100 Std.gedauert hätte. Der Grund hierfür liegt in der Architektur von GPUs. GPUs haben deutlich mehr Kerne als CPUs. Diese haben meistens zwischen 4 und 16 Kerne, während GPUs mehrere Hundert bis mehrere tausend Kerne besitzen [37]. Zwar besitzen CPUs eine höhere Taktung je Kern als GPUs, jedoch profitieren KNN vor allem durch die hohe Anzahl an Kernen, welche ein großes Maß an Parallelität erlauben [37]. Diese Parallelität ist entscheidend für die bessere Performance von GPUs, da beim NLP, sowie auch allgemein bei ML-Prozessen, viele Vektor- und Matrizen-Operationen parallel durchgeführt werden können. Diese benötigen verhältnismäßig geringe Rechenleistung, wodurch die geringe Taktung der GPUs kein Problem darstellt. Generell lässt sich daraus schließen, dass GPUs gut darin sind viele kleine Probleme zu lösen und CPUs gut darin sind einzelne große Probleme zu lösen [38].

5.7 BIONDA NER vs. KNN NER

Dieser Abschnitt diskutiert die Ergebnisse aus Kapitel 4.3 in Hinblick auf die Unterschiede zwischen BIONDA und den in dieser Arbeit verwendeten Modellen.

Während die Modelle des *machine-learning approach* in allen Metriken Werte über 80 % erreichten, erreichte BIONDA nur für die Precision auf dem Disease-Dataset einen Wert über 80 % (Tabelle 4.4). Zwar ist die Precision in dem Kontext dieser Arbeit ein wichtiger Indikator, jedoch erreicht BIONDA nur auf dem Disease-Datensatz einen solch hohen Wert. Auf dem Protein/Gene-Datensatz ist die Precision von BIONDA deutlich niedriger als die der Modelle (Tabelle 4.5). Dies ist aber bedeutend, da eine Krankheit in der Regel mehrere Protein-/Gene-Biomarker hat. Das vorhanden sein einer hohe Anzahl an Biomarkern ist

daher wichtige Grundlage zur Spezifikation der Krankheit.

BIONDA erkennt zwar die meisten Krankheiten richtig, aber nur etwas mehr als die Hälfte der Proteine/Gene. Dies ist ein Nachteil des *dictionary-approach*, den BIONDA nutzt, da dieser nur die in ihm enthaltenen Gene/Proteine kennt. Eine Konsequenz hiervon ist, dass nur eine geringe Anzahl nicht besonders präziser Biomarkerkandidaten zur Verfügung steht.

Währenddessen erkennen die Modelle über 80 % aller Proteine/Gene in dem Testset richtig. Abgesehen von der Precision taggen die Modelle die Entitäten auch mit einem um bis zu über 70 Prozentpunkte höheren Recall, wodurch sie deutlich mehr Ergebnisse liefern. Dies ist ein Vorteil des *machine-learning approach*, da dieser deutlich mehr Gene/Proteine erkennt.

Die hohe Precision in Verbindung mit dem hohen Recall sorgt dafür, dass die Modelle des *machine-learning approach* eine Vielzahl an zuverlässigen Ergebnissen liefern. Zwar werden für die BIONDA DB eine hohe Anzahl an Abstracts verarbeitet, wodurch die Wahrscheinlichkeit steigt, dass auch mithilfe des *dictionary-approach* viele Biomarker erkannt werden, jedoch können so trotzdem keine Biomarker erkannt werden, die nicht in dem *dictionary* enthalten sind. Auch hilft die höhere Anzahl an Ergebnissen des *machine-learning approach* dabei ein Scoring zu erstellen, wenn z. B. berechnet werden soll, wie oft ein Biomarker mit einer Krankheit zusammen auftritt.

Kapitel 6

AUSBLICK

In dieser Arbeit wurde gezeigt, dass mittels des *machine-learning* Ansatzes Verbesserungen der Metriken (Precision, Recall und F1-Score) im Textmining erreicht werden können. Es wäre zukünftig möglich diese Ergebnisse noch weiter zu verbessern, wenn die Parameter für das *fine-tuning* individuell an jedes Modell und jeden Datensatz angepasst und optimiert werden, anstatt ein vordefiniertes Skript hierfür zu verwenden.

Um die Performance im Vergleich zu BIONDA noch besser abschätzen zu können würde sich ein Test der verschiedenen Modelle mit festen Parametern auf einer vergleichbaren Plattform anbieten. Solch ein Test könnte z. B. in der de.NBI Cloud auf einer VM mit fester Anzahl CPUs, RAM etc. durchgeführt werden. Hierfür könnte jedes Modell die gleichen 200 oder 1000 Abstracts zur NER bekommen. Für diese Aufgabe werden die Zeit und die RAM-Auslastung gemessen. Dadurch ergäbe sich dann ein weiterer objektiver Vergleich neben den in dieser Arbeit bereits vorgestellten Metriken. Anhand dieser Werte könnten dann die Modelle sowie BIONDA weiter verglichen und bewertet werden.

Des Weiteren besteht die Möglichkeit noch weitere Methoden des Textminings zu implementieren wie RE und NEN. Dies würde die in Kapitel 1.4 angesprochene Problematik mit uneinheitlichen Bezeichnungen z. B. bei medizinischen Wirkstoffen lösen und so zu einer besseren Nutzbarkeit beitragen, indem die Entitäten unter einem Begriff zusammengefasst werden können. Darauf aufbauend könnte die Implementation von RE-Methoden zu einem besseren Verständnis von Biomarker-Krankheits-Beziehungen führen und so mehr true positive Treffer erzeugen, was BIONDA interessanter für Forscher machen würde. Zudem besteht die Möglichkeit anstatt Abstracts die *full text article* von PubMed zu verwenden und hierdurch einen größeren Umfang an Informationen zu erhalten. Dadurch wäre aber auch die Gefahr auf mehr false positive Treffer erhöht. Auch durch immer neue oder verbesserte vor-

trainierte Modelle für das Textmining in der Biomedizin sind hier in Zukunft noch bessere Ergebnisse zu erwarten.

In dem konkreten Fall dieser Arbeit mit BIONDA würde sich auch die Implementierung einer Hybridstrategie anbieten. Hierbei könnten KNN-Modelle zur NER genutzt werden. Diese würden von den hohen Recall- und Precision-Werten der Modelle profitieren, während ein *dictionary approach* zur NEN eingesetzt werden könnte. Dieser würde die getaggten Entitäten mit Hilfe probabilistischer Methoden auf die Einträge des *dictionary* mappen und somit normalisieren.

Kapitel 7

FAZIT

Diese Arbeit hat gezeigt, dass vortrainierte Modelle mit nur wenig *fine-tuning* hohe Ergebnisse erzielen können. Die Ergebnisse zeigen eine klare Verbesserung bei der Erkennung von Biomarkern und Krankheiten. Durch Textmining mittels *machine-learning* Ansatz lassen sich Entitäten nicht nur mit ähnlicher oder besserer Präzision erkennen, sondern auch in deutlich größerer Anzahl als bei einem *dictionary* Ansatz. Die Einbindung solcher Modelle, in Verbindung mit RE und NEN, würde einen großen Mehrwert für BIONDA bieten, indem BIONDA mehr relevante Informationen zur Verfügung stünden.

Abkürzungsverzeichnis

ELISA Enzyme-Linked Immunosorbent Assay. 1

MS Massenspektrometrie. 1

DB Datenbank. 1

ML machine-learnings. 1

NLP Natural Language Processing. 1

NER Named Entity Recognition. 1

DNA Desoxyribonukleinsäure. 6

RNA Ribonukleinsäuren. 6

mRNA messenger RNA. 6

tRNA transfer RNA. 7

miRNA microRNA. 7

WHO World Health Organisation. 7

HbA1c Hämoglobin A1c. 8

DM Diabetes mellitus. 8

TM Textmining. 8

KNN künstliches neuronales Netz. 8

NEN Namend Entity Normalization. 8

RE Relation Extraction. 8

MeSH Medical Subject Headings. 9

LSTM Long short-term memory. 12

CRF Conditional Random Fields. 12

CNN Convolutional Neural Networks. 16

VM Virtual Machine. 17

GPU Graphics Processing Unit. 31

VCPU Virtuell Central Processing Units. 31

Literaturverzeichnis

[1] WHO. *Environmental Health Criteria 222 Biomarkers In Risk Assessment: Validity And Validation*. 2001. URL: http://www.inchem.org/documents/ehc/ehc/ehc222.htm.

[2] FDA-NIH Biomarker Working Group. *Diagnostic Biomarker*. Silver Spring (MD): Food und Drug Administration (US) Co-published by National Institutes of Health (US), Bethesda (MD), 2016. URL: https://www.ncbi.nlm.nih.gov/books/NBK402285/.

[3] FDA-NIH Biomarker Working Group. *Understanding Prognostic versus Predictive Biomarkers*. Silver Spring (MD): Food und Drug Administration (US) Co-published by National Institutes of Health (US), Bethesda (MD), 2016. URL: https://www.ncbi.nlm.nih.gov/books/NBK402284/.

[4] Lippman ME Yang X. *BRCA1 and BRCA2 in breast cancer*. DOI: 10.1023/a:100618990689.

[5] Xuan Wang u. a. "Cross-type Biomedical Named Entity Recognition with Deep Multi-Task Learning". In: *Bioinformatics (Oxford, England)* 35 (Jan. 2018). DOI: 10.1093/bioinformatics/bty869.

[6] Griffiths AJF u. a. *An Introduction to Genetic Analysis*. 7. Aufl. New York: W. H. Freeman, 2000. URL: https://www.ncbi.nlm.nih.gov/books/NBK21811/.

[7] Vernon M. Ingram. "Sickle-Cell Anemia Hemoglobin: The Molecular Biology of the First "Molecular Disease"—The Crucial Importance of Serendipity". In: *Genetics* 167.1 (2004), S. 1–7. ISSN: 0016-6731. DOI: 10.1534/genetics.167.1.1. eprint: https://www.genetics.org/content/167/1/1.full.pdf. URL: https://www.genetics.org/content/167/1/1.

[8] Aaron David Goldman und Laura F Landweber. "What Is a Genome?" In: *PLoS genetics* 12.7 (2016). ISSN: 1553-7390. DOI: 10.1371/journal.pgen.1006181. eprint: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4956268/.

[9] Francis S Collins und Leslie Fink. "The Human Genome Project". In: *Alcohol health and research world* 19 (1995), S. 190–195. ISSN: 0090-838X. eprint: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6875757/.

[10] Rosa Visone und Carlo M. Croce. "MiRNAs and Cancer". In: *The American Journal of Pathology* 174.4 (2009), S. 1131–1138. ISSN: 0002-9440. DOI: https://doi.org/10.2353/ajpath.2009.080794. URL: http://www.sciencedirect.com/science/article/pii/S0002944010609728.

[11] Jacob O'Brien u. a. "Overview of MicroRNA Biogenesis, Mechanisms of Actions, and Circulation". In: *Frontiers in endocrinology* 9 (2018), S. 402. ISSN: 1664-2392. DOI: 10.3389/fendo.2018.00402. eprint: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6085463/.

[12] David Gonzalez u. a. "Mutational Status of the TP53 Gene As a Predictor of Response and Survival in Patients With Chronic Lymphocytic Leukemia: Results From the LRF CLL4 Trial". In: *Journal of Clinical Oncology* 29.16 (2011). PMID: 21483000, S. 2223–2229. DOI: 10.1200/JCO.2010.32.0838. eprint: https://doi.org/10.1200/JCO.2010.32.0838. URL: https://doi.org/10.1200/JCO.2010.32.0838.

[13] MV Relling u. a. "Clinical Pharmacogenetics Implementation Consortium Guidelines for Thiopurine Methyltransferase Genotype and Thiopurine Dosing". In: *Clinical Pharmacology & Therapeutics* 89.3 (2011), S. 387–391. DOI: https://doi.org/10.1038/clpt.2010.320. eprint: https://ascpt.onlinelibrary.wiley.com/doi/pdf/10.1038/clpt.2010.320. URL: https://ascpt.onlinelibrary.wiley.com/doi/abs/10.1038/clpt.2010.320.

[14] Nosheen Mahmood u. a. "Circulating miR-21 as a prognostic and predictive biomarker in oral squamous cell carcinoma". In: *Pakistan Journal of Medical Sciences* 35 (Aug. 2019). DOI: 10.12669/pjms.35.5.331.

[15] Fei Zhu u. a. "Biomedical text mining and its applications in cancer research". In: *Journal of biomedical informatics* 46 (Nov. 2012). DOI: 10.1016/j.jbi.2012.10.007.

[16] Hoehndorf R. Rebholz-Schuhmann D Oellrich A. "Text-mining solutions for biomedical research: enabling integrative biology". In: *Nat Rev Genet* 13 (2012). DOI: 10.1038/nrg3337.

[17] Lee J. E. "How Should Biobanks Collect Biosamples for Clinical Application? A 20-year Biomarker-related Publication and Patent Trend Analysis". In: *Osong public health and research perspectives* 9.3 (2018), S. 105–111. DOI: https://doi.org/10.24171/j.phrp.2018.9.3.04.

[18] Jinhyuk Lee u. a. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* (Sep. 2019). ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz682. URL: https://doi.org/10.1093/bioinformatics/btz682.

[19] Yu Gu u. a. *Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing*. 2020. eprint: arXiv:2007.15779.

[20] Peng Qi u. a. "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, Juli 2020, S. 101–108. DOI: 10.18653/v1/2020.acl-demos.14. URL: https://www.aclweb.org/anthology/2020.acl-demos.14.

[21] Mark Neumann u. a. "ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing". In: *Proceedings of the 18th BioNLP Workshop and Shared Task*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, S. 319–327. DOI: 10.18653/v1/W19-5034. eprint: arXiv:1902.07669. URL: https://www.aclweb.org/anthology/W19-5034.

[22] Lu Z. Doğan RI Leaman R. "NCBI disease corpus: a resource for disease name recognition and concept normalization". In: *J Biomed Inform* 47 (2014). DOI: 10.1016/j.jbi.2013.12.006..

[23] Ming-Siang Huang u. a. "Biomedical named entity recognition and linking datasets: survey and our recent development". In: *Briefings in Bioinformatics* 21.6 (Juni 2020), S. 2219–2238. ISSN: 1477-4054. DOI: 10.1093/bib/bbaa054. URL: http://dx.doi.org/10.1093/bib/bbaa054.

[24] Juliane Fluck u. a. "Information Extraction Technologies for the Life Science Industry". In: *Drug Discovery Today: Technologies* 2 (Sep. 2005), S. 217–224. DOI: 10.1016/j.ddtec.2005.08.013.

[25] Leon Weber u. a. "HunFlair: An Easy-to-Use Tool for State-of-the-Art Biomedical Named Entity Recognition". In: *arXiv preprint arXiv:2008.07347* (2020).

[26] Leon Weber u. a. "HUNER: improving biomedical NER with pretraining". In: *Bioinformatics (Oxford, England)* 36.1 (Jan. 2020), S. 295–302. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz528. URL: https://doi.org/10.1093/bioinformatics/btz528.

[27] Alan Akbik, Duncan Blythe und Roland Vollgraf. "Contextual String Embeddings for Sequence Labeling". In: *COLING 2018, 27th International Conference on Computational Linguistics*. 2018, S. 1638–1649.

[28] Alan Akbik, Tanja Bergmann und Roland Vollgraf. "Pooled Contextualized Embeddings for Named Entity Recognition". In: *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 2019, S. 724–728.

[29] Charles Sutton und A. McCallum. "An Introduction to Conditional Random Fields". In: *Found. Trends Mach. Learn.* 4 (2012), S. 267–373.

[30] Piotr Bojanowski u. a. "Enriching Word Vectors with Subword Information". In: *arXiv preprint arXiv:1607.04606* (2016).

[31] Jacob Devlin u. a. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Juni 2019, S. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://www.aclweb.org/anthology/N19-1423.

[32] Ashish Vaswani u. a. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[33] Sunil Mohan und Donghui Li. "MedMentions: A Large Biomedical Corpus Annotated with UMLS Concepts". In: (Mai 2019). URL: https://arxiv.org/abs/1902.09476.

[34] Guillaume Lample u. a. "Neural Architectures for Named Entity Recognition". In: *Proc. NAACL-HLT*. 2016.

[35] L. Ramshaw und M. Marcus. "Text Chunking using Transformation-Based Learning". In: *ArXiv* cmp-lg/9505040 (1995).

[36] Naomi Most. Version 0.5.5. URL: https://pypi.org/project/metapub/.

[37] Shubham Gupta und M.R. Babu. "Performance analysis of gpu compared to single-core and multi-core cpu for natural language applications". In: *International Journal of Advanced Computer Science and Applications* 2 (Juni 2011), S. 50–53.

[38] Yihui Ren, Shinjae Yoo und Adolfy Hoisie. "Performance Analysis of Deep Learning Workloads on Leading-edge Systems". In: *CoRR* abs/1905.08764 (2019). arXiv: 1905.08764. URL: http://arxiv.org/abs/1905.08764.

Abbildungsverzeichnis

1.1 **Anzahl der Ergebnisse für die Suchanfrage "biological marker[MeSH Terms] & Human[organism]" in PubMed, für die Jahre 1946 bis 2021.** Die Y-Achse zeigt die jährliche Anzahl von Veröffentlichungen und die X-Achse die Jahre 1946 bis 2021. Bis 1985 steigt der Graph nur langsam an, danach steigt er deutlich stärker. Die Anzahl an Veröffentlichungen erreicht ihren Höhepunkt im Jahr 2019 mit 49255 [17]. 9

1.2 **Phasen und Aufgaben des Biomedizinischen Textminings mit KNN [15]** 10

2.1 **HunFlair pretraining [25].** Es zeigt die Architektur von HunFlair. Der erste Teil basiert auf einem Sprachmodell auf Flair-Zeichenebene, das auf rund 24 Millionen Abstracts biomedizinischer Artikel von PubMed und 3 Millionen Volltexten aus PMC sowie auf FastText-Worteinbettungen trainiert wurde [30]. Der zweite Teil, die Vorhersage benannter Entitäten, wird von einem BiLSTMCRF-Modell durchgeführt. Analog zu HUNNER werden für jeden Entitätstyp unterschiedliche Modelle trainiert, wobei alle Trainingssätze aller Goldstandard-NER-Korpora mit diesem Typ kombiniert werden, um die Leistung gegenüber Textgenres und biomedizinischen Subdomänen zu verbessern. 13

2.2 **Mixed-Domain Pretraining von BioBERT [18].** Zum Vortraining wird BioBERT mit den Parametern von BERT initialisiert, die in Wikipedia trainiert wurden. Von diesem Modell wird ein Pretraining mit Texten durchgeführt, die aus PubMed und PMC extrahiert wurden. 14

2.3 **Fine-tuning von BioBERT.** In BioBERT können unterschiedliche Aufgaben *fine-tuned* werden. In dieser Arbeit werden wir uns auf NER konzentrieren. [18] 15

2.4 **Mixed-Domain Pretraining:** Das Pretraining wird erst auf einem Wortschatz aus z. B. Wikipedia und Büchern durchgeführt und anschließend auf Biomedizinischen Texten z. B. aus PubMed [19] 15

2.5 **PubMedBERT pretraining:** Das Pretraining wird auschließlich auf einem Wortschatz aus biomedizinschen Texten z. B. PubMed Abstracts durchgeführt [19] 16

3.1 **Aufbau der TSV Datei.** Beispielhaft an dem NCBI-disease Datensatz . . 19

Tabellenverzeichnis

2.1 **Goldstandarddatensätze [5]** 17

4.1 **Formeln für Precision, Recall und F1-Score** 23
4.2 **Definition von True Positive, True Negative, False Positive und False Negative für das BIO Format.** Bei einem *Multiword Token* wird ein Treffer bereits als True Positiv gezählt, wenn nur ein Teil der Entität erkannt wurde. 24
4.3 **Macro-Ergebnisse der Modelle.** Ergebnisse der Modelle BioBERT, Hunflair und PubMedBERT mit den Parametern Precision, Recall und F1-Score, nach dem Training auf den unter Datasets angegebenen Goldstandarddatensätzen. 25
4.4 **Ergebnisse der Modelle für Precision, Recall und F1-Score nach dem Training auf den kombinierten Datensätzen für den Entitätentyp Diseases** 26
4.5 **Ergebnisse der Modelle für Precision, Recall und F1-Score nach dem Training auf den kombinierten Datensätzen für den Entitätentyp Proteine/Gene** 26

5.1 **F1-Score von PubMedBERT aus der Publikation [19] und dieser Arbeit für NER auf den Testteilen der verschiedenen Datensätze** . 29
5.2 **F1-Score von BioBERT aus der Publikation [19] und dieser Arbeit für NER auf den Testteilen der verschiedenen Datensätze** 29