

# NLP zur Unterstützung von SNOMED CT Codierung

SNOMAST 2.0, ein Prototyp für SNOMED CT Codierungen von ärztlichen Weiterbildungstexten in deutscher Sprache

Bachelorthesis

Studiengang:

Autoren:

Betreuer:

Auftraggeber:

Experte:

Datum:

Medizininformatik

Sebastian Kunz und Cyril Zraggen

Prof. Dr. Murat Sariyar

SIWF Schweizerisches Institut für ärztliche Weiter- und Fortbildung

Reto Mettler

16.06.2022

## Management Summary

Täglich werden immer mehr digitale Daten generiert, gespeichert und bearbeitet. Dabei liegen im Gesundheitswesen rund 80 % in einer nicht strukturierten und nicht maschinenlesbaren Form vor. Also meist Text und das ist eine Herausforderung. Nicht angemessener Umgang mit diesen Daten kann zum Verlust von medizinisch-relevanten Informationen führen. Natural Language Processing (NLP) kann bei der maschinellen Verarbeitung dieser Daten unterstützen indem es hilft Informationen aus Texten zu extrahieren und zu strukturieren. Unsere Motivation als angehende Medizininformatiker war und ist es, smarte technische Lösungen zusammen mit den Stakeholdern im Gesundheitswesen, zu entwickeln, die eine zweck- und patientenorientierte Bearbeitung ermöglichen. Diese Arbeit zeigt anhand eines Anwendungsfalls wie eine solche Lösung erarbeitet werden kann.

Das Ziel dieser Arbeit war es, das Schweizerische Institut für ärztliche Weiter- und Fortbildung beim Mapping zwischen textuellen Beschreibungen aus den Weiterbildungsprogrammen und SNOMED CT zu unterstützen. Hierfür haben wir die Desktop Applikation SNOMAST 2.0 für die Verwendung von Bidirectional Encoder Representations from Transformers (BERT) entwickelt. BERT wurde 2018 von Google veröffentlicht und erzielt noch vier Jahre später in NLP-Aufgaben state-of-the-art Ergebnisse.

Wir verwendeten eine Weiterentwicklung von BERT namens BioBERT. Ein BERT-Modell, das speziell auf biomedizinischen Texten trainiert wurde und bei biomedizinischen Texten bessere Ergebnisse als BERT erzielt. Insgesamt wurden drei BioBERT-Modelle weiterentwickelt, die von SNOMAST 2.0 verwendet werden können. Die Modelle berechnen die wahrscheinlichsten SNOMED CT Konzepte, die in einem Text enthalten sein können. Unsere Modelle können je nach Modell und entsprechendem Klassifikator – zwischen 2 bis 89'855 Konzepten unterscheiden.

Wir erzielten in der Evaluation über den gesamten Trainingsdatensatz, bei allen drei Modellen eine Gesamtgenauigkeit (F1-Wert) von über 70 %. Unser erstes Hauptziel, «Einen Prototyp für das Entity-Linking auf SNOMED CT über NLP auf Basis von BERT zu erstellen und einen F1-Wert über 70 % zu erreichen», haben wir somit erreicht. Zudem wurde durch das Schweizerische Institut für ärztliche Weiter- und Fortbildung der Prototyp «SNOMAST 2.0» mit einem ausgezeichneten System Usability Score von 88.75 bewertet. Dank dem SUS Score kann die Benutzerfreundlichkeit einer Applikation erfasst werden. Damit ist auch das zweite Hauptziel, «Der Prototyp erzielt bei der Gesamtevaluation nach der Abnahme durch das SIWF einen System Usability Score von mindestens 68 Punkten», erreicht worden.

Obwohl die Vorbereitung des Trainingsdatensatzes aufwendig war, die Trainingsdauer mit steigender Datensatzgrösse zunimmt und wir einen zu geringen Trainingsdatensatz verwendeten, gelang es uns mit den überwacht trainierten Modellen akzeptable Resultate zu erzielen. Darüber hinaus zeigte die Gesamtevaluation, dass SNOMAST 2.0 das Schweizerische Institut für ärztliche Weiter- und Fortbildung beim Mapping unterstützen kann. Besser noch, es sind Bedürfnisse für neue Anwendungsfälle bei unserem Industriepartner geweckt worden.

Dank der Skalierbarkeit des Klassifikators können weitere Modelle für neue Anwendungsfälle trainiert und anschliessend in SNOMAST 2.0 verwendet werden. Somit könnten Modelle entworfen werden, die bei der Leistungscodierung aus Verlaufsberichten helfen. Die Metadaten anhand des Textes im Dokument beim Hochladen in das elektronische Patientendossier automatisch befüllen. Oder helfen die richtigen ICD-10 Codes aus medizinischen Dokumentationen zu erkennen.

### **Sprachliche Gleichstellung:**

Für die Erarbeitung dieses Dokuments haben wir uns an dem Leitfaden für die sprachliche Gleichstellung der BFH vom März 2014 orientiert. Es wurde darauf geachtet, wo immer möglich, die neutrale Form zu verwenden, um Personen aller Geschlechter mit einzubeziehen.

## Danksagung

Wir möchten uns bei allen Beteiligten bedanken, die uns bei dieser Bachelor-Thesis unterstützt haben. Ein besonderer Dank geht an:

- Unseren Betreuer Prof. Dr. Murat Sariyar, der uns stets mit seinen Fachkenntnissen gefordert und gefördert hat, beratend zur Seite stand und massgeblich unser Interesse in Natural Language Processing und BERT gesteigert und manifestiert hat.
- An unsere Auftraggeber und Projektpartner vom Schweizerischen Institut für ärztliche Weiter- und Fortbildung Lukas Wyss und Philip Kyburz, für die unkomplizierte, wertschätzende und flexible Zusammenarbeit sowie den vielen nützlichen Feedbacks und Tipps.
- An Peter von Niederhäusern für die Einführung und Support bei der Nutzung der Serverinfrastruktur der BFH.
- An alle Korrektorinnen und Korrektoren dieser Arbeit.
- An unseren Experten Reto Mettler für seine unterstützenden Rückmeldungen und Tipps beim Kick-off Meeting.
- An unsere Familien, die uns während des gesamten Studiums und ganz besonders im Laufe dieser Arbeit trugen.
- Und natürlich an BERT für seinen Lernwillen, trotz zeitweiliger Lernschwäche.

# Inhaltsverzeichnis

1	Einleitung	5
1.1	Ausgangslage	5
1.2	Anwendungszweck SNOMAST 2.0	7
1.3	BERT	8
1.4	Ziele und Fragestellungen	9
1.5	Abgrenzung	10
2	Grundlagen	11
2.1	Natural Language Processing	11
2.1.1	Word Embeddings	12
2.1.2	NLP-Modell	13
2.1.3	Neuronale Netze	14
2.2	Transfer Learning	15
2.2.1	Pretraining	15
2.2.2	Finetuning	15
2.3	BERT, Transformers und Attention	17
2.4	SNOMED CT	19
3	Methodik	20
3.1	Projektmanagement	20
3.1.1	Organisation	20
3.1.2	Risikoanalyse	21
3.1.3	Planung	21
3.2	Anforderungsanalyse	21
3.3	Recherche	22
3.4	Konzept	22
3.4.1	Grobkonzept	23
3.4.2	Detaillkonzept	25
3.4.2.1	Technologieentscheid	26
3.4.2.2	Bereich Finetuning	27
3.4.2.3	Bereich Umsetzung	31
3.4.2.4	Bereich Gesamtevaluation	31
4	Ergebnisse	35
4.1	Technologieentscheid	35
4.2	Bereich Finetuning	37
4.2.1	Modell	38
4.2.2	Validation F1 und Test F1	38
4.2.3	Training loss und Validation loss	40
4.3	Bereich Umsetzung	42
4.3.1	Prototyp	43
4.3.2	GUI	44
4.4	Bereich Gesamtevaluation	47
4.4.1	Leistungsmerkmale und Metriken	47
4.4.2	SUS Score	49
4.5	Zusammenfassung Ergebnisse	50
5	Diskussion	51
5.1	Zielerreichung Hauptziel 1	52
5.2	Zielerreichung Hauptziel 2	55

5.3 Erkenntnisse	57
5.4 Fazit	60
5.5 Ausblick	61
6 Abbildungsverzeichnis	63
7 Tabellenverzeichnis	64
8 Glossar	65
9 Literatur	70
10 Anhang	72

# 1 Einleitung

Der Anteil digital generierter Informationen nimmt rasant zu. Das Unternehmen Statista prognostiziert, dass sich das Datenvolumen von 2018 bis 2025 auf 175 Zettabyte mehr als verfünffachen wird (1). Dies entspricht einer jährlichen Zunahme von ungefähr 20 Billionen Gigabyte. Auch im Gesundheitswesen wird das Datenvolumen entsprechend zunehmen. Im Gesundheitswesen liegen 80 % der Informationen in einer unstrukturierten und nicht maschinenlesbaren Form vor (2). In Anbetracht der Zunahme des Datenvolumens und dem Wissen, dass 4/5 der medizinischen Dokumentation freitextlich erfasst wird, braucht es technische und semantische Lösungen der Medizininformatik. Lösungen, wie die Festlegung einheitlicher Spezifikationen, die eine gemeinsame Sprache (Ontologie, Terminologien, Nomenklaturen etc.) verwenden, um Informationen in maschinenlesbare Konzepte zu codieren sowie den Einsatz von Technologien, die geschriebene oder gesprochene Sprache maschinenlesbar erfassen.

Einen vermehrten und vor allem automatisierten Einsatz von Nomenklaturen könnte in Zukunft für die maschinelle Erfassung medizinisch-relevanter Informationen hilfreich sein. Denn damit können klinische Informationen sprachunabhängig als Konzepte repräsentiert werden. Eine der am weitesten und umfassendsten ontologiebasierten Nomenklaturen für den medizinischen Bereich ist die Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) (3). Die Technologie, um Sprache maschinell verarbeiten zu können, ist Natural Language Processing (NLP).

Systeme, die NLP anwenden, haben sich von handcodierten und regelbasierten Programmen wie beispielsweise ELIZA von Joseph Weizenbaum in den Sechzigerjahren (4), zu Systemen die Deep Learning und künstliche neuronale Netze anwenden, weiterentwickelt (5). Einen Meilenstein für die breite Anwendung von NLP wurde 2018 durch Google mit der Veröffentlichung von BERT (Bidirectional Encoder Representations from Transformers) gelegt (6, 7).

Im Gegensatz zu früheren NLP-Modellen ist BERT in der Lage, Wörter im Zusammenhang mit allen anderen Wörtern in einem Satz zu verarbeiten. Es kann den gesamten Kontext eines Satzes berücksichtigen, indem es die Wörter und deren Beziehungen gleichzeitig betrachtet. Seit der Veröffentlichung von BERT gibt es einige wissenschaftliche Arbeiten, die BERT erfolgreich anwendeten, um die Erkennung von SNOMED CT Konzepten aus unstrukturiertem Text (Entity-Linking) zu ermöglichen (8–10).

Innerhalb dieser Arbeit verwendeten wir BioBERT. Das ist ein BERT-Modell, dass auf biomedizinischen Texten trainiert wurde. Ausserdem übertrifft BioBERT die bisherigen Ergebnisse von BERT bei biomedizinischen Aufgaben, beispielsweise bei der Named Entity Recognition in biomedizinischen Texten (11). Mit einem F1-Wert von 89.71 erzielte BioBERT bis heute auf der Krankheitserkennung beim «NCBI Disease» Datensatz das beste Resultat (12). Der F1-Wert F1 ist ein Leistungsmerkmal bei der Modellevaluation nach dem Training eines Classifiers und beschreibt die Gesamtgenauigkeit der Vorhersagen.

In dieser Arbeit werden wir aufzeigen wie unstrukturierte, textbasierte und medizinische Informationen aus Dokumenten, durch den Einsatz eines überwacht trainierten BERT-Modells, extrahiert und in die dazugehörigen SNOMED CT Konzepte abgebildet werden können. Anhand eines spezifischen Anwendungsfall wird dargelegt, wie der von uns entwickelte Prototyp «SNOMAST 2.0» in einem Geschäftsprozess von unserem Industriepartner dem Schweizerischen Institut für ärztliche Weiter- und Fortbildung (SIWF) unterstützend eingesetzt werden kann.

## 1.1 Ausgangslage

Das SIWF ist ein autonomes Organ des Berufsverbands der Schweizer Ärzte, der Foederatio Medicorum Helveticorum (FMH) und zentrale Anlaufstelle für die ärztliche Weiter- und Fortbildung. Das SIWF erteilt im Auftrag des Eidgenössischen Departement des Innern (EDI) jährlich mehr als tausend eidgenössische Weiterbildungstitel an Ärztinnen und Ärzte (13). Die Ärzteschaft nutzt das elektronische Logbuch vom SIWF für die Dokumentation ihrer Weiterbildungen. Im Rahmen der Weiterentwicklung des elektronischen Logbuchs wurde eine Vereinheitlichung der Anforderungsbeschreibungen, auch als Prozeduren bezeichnet, über SNOMED CT angegangen, da es immer wieder zu semantischen

Ambiguitäten bei den Einträgen kommt. Das SIWF ist ein autonomes Organ des Berufsverbands der Schweizer Ärzte, der Foederatio Medicorum Helveticorum (FMH) und zentrale Anlaufstelle für die ärztliche Weiter- und Fortbildung. Das SIWF erteilt im Auftrag des Eidgenössischen Departement des Innern (EDI) jährlich mehr als tausend eidgenössische Weiterbildungstitel an Ärztinnen und Ärzte (13). Die Ärzteschaft nutzt das elektronische Logbuch vom SIWF für die Dokumentation ihrer Weiterbildungen. Im Rahmen der Weiterentwicklung des elektronischen Logbuchs wurde eine Vereinheitlichung der Anforderungsbeschreibungen, auch als Prozeduren bezeichnet, über SNOMED CT angegangen, da es immer wieder zu semantischen Ambiguitäten bei den Einträgen kommt. Um das SIWF bei der manuellen Annotation zwischen diesen Prozeduren und SNOMED CT zu unterstützen, wurde in einer früheren Arbeit ein erster Prototyp für das Entity-Linking über NLP auf Basis eines open-source Medical Concept Annotation Toolkit (MedCAT) erstellt. Die mit MedCAT erstellten NLP Modellen lieferten jedoch ernüchternde Ergebnisse (F1 Wert <0.5). Die Ursachen für die enttäuschenden Ergebnisse lagen zum einen in einem zu geringen Trainingsdatensatz und zum anderen in der fehlenden Kontexterkenkung des Modells. Die NLP Pipelines, welche von MedCat generiert werden, beziehen den Kontext eines Wortes im Rahmen eines Sichtfensters ein. Übergeordnete Kontexte konnten so jedoch nicht mit einbezogen werden. Hier kommt BERT ins Spiel, das den gesamten Kontext eines Satzes berücksichtigt und die Wörter eines Satzes und deren Beziehungen gleichzeitig betrachtet.

## 1.2 Anwendungszweck SNOMAST 2.0

Um die Ausgangslage und den Mehrwert unserer Arbeit zu verdeutlichen, wird in diesem Kapitel der Anwendungszweck des entwickelten Prototyps «SNOMAST 2.0» erläutert. Konkret soll der Prototyp das SIWF bei der manuellen Annotation des Mappings zwischen textuellen Beschreibungen und SNOMED CT Konzepten unterstützen.

Damit ersichtlich wird, wann der Prototyp hierfür zum Einsatz kommt, wird nachfolgend der Geschäftsprozess vom SIWF mit Hilfe der Business Process Modell Notation abgebildet. Wir verzichten an dieser Stelle darauf, den Prozess vollständig zu kommentieren, dies erfolgt im Pflichtenheft (Anhang 10.5) – stattdessen wird der Prozess vereinfacht beschrieben.

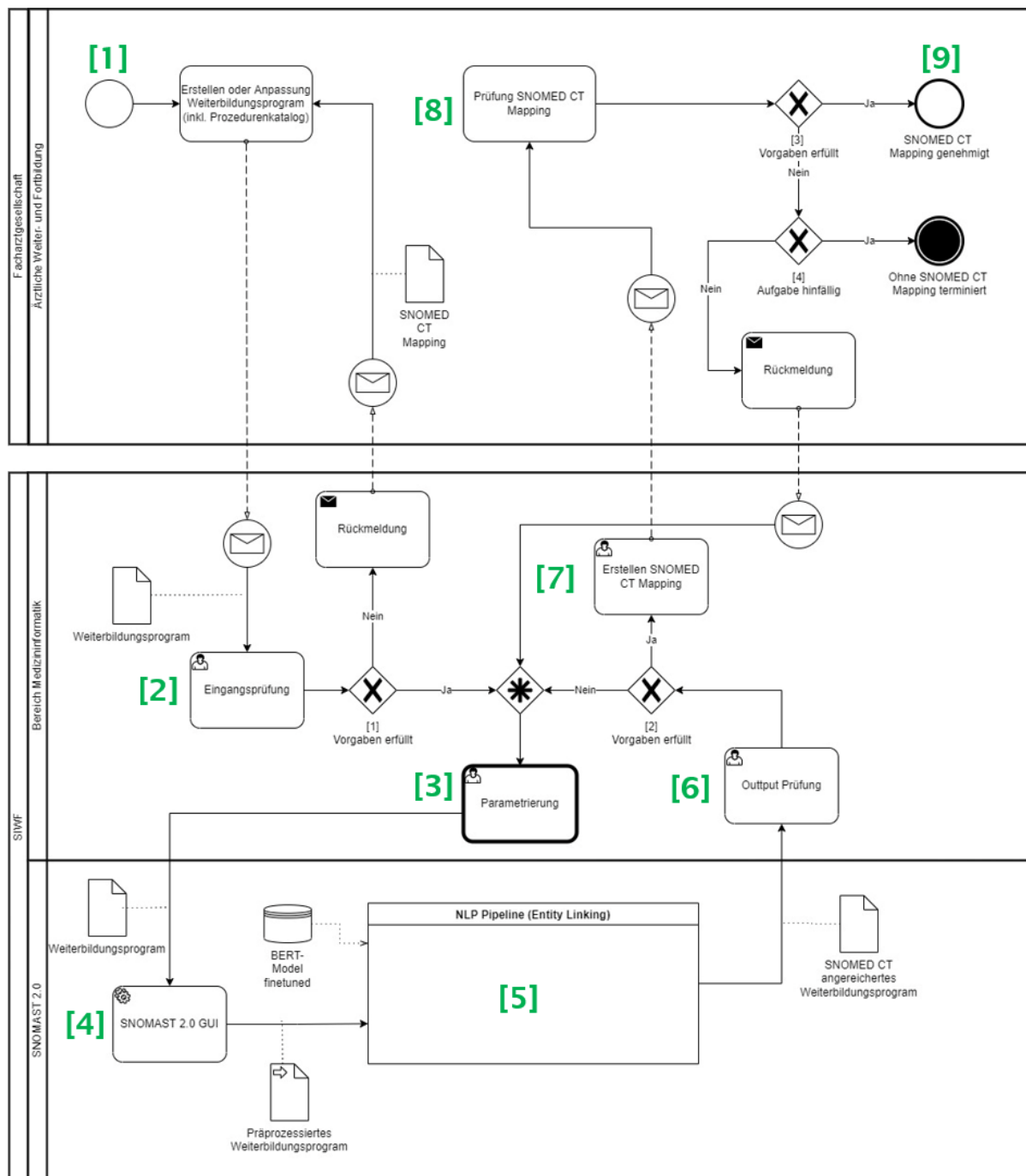


Abbildung 1: Geschäftsprozess SIWF - Anwendungszweck SNOMAST 2.0.



Der Geschäftsprozess startet mit dem Auftrag zur Erstellung oder Anpassung eines Weiterbildungsprogramms in deutscher Sprache für einen Facharzttitel durch die zuständige Facharztgesellschaft [1]. Das erstellte oder aktualisierte Weiterbildungsprogramm wird dem verantwortlichen Bereich Medizininformatik beim SIWF übergeben. Es erfolgt eine Eingangsprüfung, bei welcher u.a. kontrolliert wird, ob die enthaltenen Prozeduren so spezifisch wie möglich definiert sind [2].

Sind die Vorgaben erfüllt, führt das SIWF die Parametrierung durch [3]. Hierfür wird das Weiterbildungsprogramm via grafischer Benutzeroberfläche (engl. graphical user interface – GUI) dem NLP-Modell übergeben [4]. Der Prototyp führt mit dem erhaltenen Text und dem ausgewählten BERT-Modell das Entity-Linking durch [5]. Als Resultate werden pro Text, die wahrscheinlichsten SNOMED CT Codes als Output ausgegeben. Der Output wird durch die Medizininformatiker vom SIWF auf ihre Plausibilität hin überprüft [6].

Ist der Output schlüssig, erfolgt das «SNOMED CT Mapping», bei dem der angegebene SNOMED CT Code der entsprechenden Prozedur zugewiesen wird. Es können auch mehrere Codes pro Prozedur verwendet werden. Die Korrektheit der erstellten Annotationen wird von der Fachgesellschaft geprüft [8] und durch eine Unterschrift der zuständigen Person bestätigt. Der Prozess endet weitgehend mit dem Ergebnis als abgenommener Prozeduren Katalog des Facharzttitels [9].

### 1.3 BERT

BERT ist ein NLP-Modell, das auf einen riesigen Datensatz aus englischen Wikipedia Einträgen (2'500 Mio. Wörter) sowie Einträgen aus dem BookCorpus (800 Mio. Wörter) vortrainiert wurde (14). BERT ist öffentlich für jedermann zugänglich und im Gegensatz zu den bisherigen NLP-Modellen in der Lage, den vor- und nachgelagerten Kontext eines Wortes zu berücksichtigen. BERT ist ein NLP-Modell, das auf einen riesigen Datensatz aus englischen Wikipedia Einträgen (2'500 Mio. Wörter) sowie Einträgen aus dem BookCorpus (800 Mio. Wörter) vortrainiert wurde (14). BERT ist öffentlich für jedermann zugänglich und im Gegensatz zu den bisherigen NLP-Modellen in der Lage, den vor- und nachgelagerten Kontext eines Wortes zu berücksichtigen. BERT erzielt in einer Vielzahl von NLP Aufgaben state-of-the-art Ergebnisse (6), beispielsweise in NLP Aufgaben wie «Question Answering», «Text Classification» oder «Word-sense disambiguation» (Auflösung von Ambiguitäten) und begegnet uns täglich bei der Benutzung der Suchmaschine von Google (15).

Die guten Ergebnisse sowie der Open-source Ansatz führte dazu, dass es inzwischen eine Vielzahl von frei verfügbaren und weiterentwickelten BERT-Modellen für unterschiedliche Domänen gibt, bspw. BioBERT für Biomedizin (11), GottBERT für die deutsche Sprache (16) oder FinBERT für den Finanzsektor (17). Weil BERT heute state-of-the-art Ergebnisse in der Textklassifizierung erzielt, frei verfügbar und bereits auf einen riesigen Datensatz vortrainiert wurde, wird in dieser Arbeit das bestehende BioBERT Modell mittels Transfer Learning verwendet, um einen Klassifizierer mit einem künstlich neuronalen Netz (Finetuning) erweitert und auf den aufgeführten Anwendungsfall angewendet. Wie das zugrundeliegende BERT Modell konzipiert ist, was dessen Konzepte sind und wie Transfer Learning und Finetuning konzeptionell funktionieren, wird im Kapitel Grundlagen erläutert.

## 1.4 Ziele und Fragestellungen

Für diese Bachelorarbeit ist folgendes Ziel durch den Betreuer in der Aufgabenstellung an die Studierenden formuliert worden (Anhang 11.1):

*Ziel dieser Bachelorarbeit ist die Verbesserung des Entity-Linkings in dem bisherigen Prototyp über vier Maßnahmen. Erstens soll von Word2Vec auf BERT bzw. BioBERT umgestiegen, also ein dynamisches Word Embedding anvisiert werden. Für das zum Einsatz kommende Self-supervised Learning sollen dabei auch Leitlinien von der AWMF genutzt werden. Zweitens sind für die Word Embeddings ein Finetuning, Gütekriterien und Visualisierungsmöglichkeiten zu entwickeln, auch damit der Stakeholder versteht, was es mit diesen Embeddings auf sich hat. Drittens ist zu untersuchen, ob bereits ein Teil von SNOMED CT für das Mapping ausreicht (Stichwort Ontologie-Extraktion). Schließlich soll viertens eine Lösung für Probleme bei der Übersetzung ins Englische gefunden werden, z.B. über das Nutzen von <https://huggingface.co/bertbase-german-cased> und damit Überflüssigmachen der Übersetzung.*

*Insbesondere sind folgende Punkte anzugehen:*

- *dynamische Word Embeddings lernen, finetunen und analysieren*
- *BERT für die deutsche klinische Sprache nutzen*
- *Gesamtlösung für das Entity-Linking implementieren, das F1-Werte über 70% erzielt*

Aus der genannten Aufgabenstellung und nach einer Abgrenzung (Kapitel 1.5) wurden zwei Hauptziele (1.HZ & 2.HZ) für die BSc. Thesis nach SMART-Kriterien definiert. Daraus leiten sich die Fragestellungen (1.1F bis 2.2F) für die BSc. Thesis und die technischen Ziele im Pflichtenheft (Anhang 10.5) ab.

ID	Typ	Beschreibung
1.HZ	Hauptziel	Ein Prototyp für das Entity-Linking auf SNOMED CT über NLP auf Basis von BERT ist am Ende der BSc. Thesis erstellt und erzielt einen F1-Wert über 70 %.
1.1F	Fragestellung	Welche Massnahmen führen dazu, dass das Entity-Linking gegenüber dem vorherigen Prototyp aus der Living Case 2 Arbeit verbessert wird?
1.2F	Fragestellung	Wie sieht die Gesamtarchitektur aus, damit der Prototyp einfach auf neue Bedürfnisse hin angepasst werden kann?
1.3F	Fragestellung	Wie kann mit Transfer Learning das gewählte BERT-Modell einen F1-Wert über 70 % erzielen?
2.HZ	Hauptziel	Der Prototyp erzielt bei der Gesamtevaluation nach der Abnahme einen System Usability Score von mindestens 68 beim SIWF.
2.1F	Fragestellung	Welche Funktionen muss der Prototyp zur Verfügung stellen, damit das SIWF bei der manuellen Annotation des Mappings zwischen textuellen Beschreibungen und SNOMED CT unterstützt werden kann?
2.2F	Fragestellung	Was sind geeignete Visualisierungsmöglichkeiten vom Word Embedding, damit das SIWF bei der Einschätzung des Modells unterstützt wird?

Tabelle 1: Übersicht Hauptziele und Fragestellungen

## 1.5 Abgrenzung

Die aufgezeigten Ziele und Fragestellungen unterscheiden sich von den in der Aufgabenstellung formulierten Massnahmen. In diesem Kapitel werden die festgelegten Abgrenzungen innerhalb dieser Bachelorarbeit aufgeführt und mit Fragen und dazugehörigen Antworten begründet.

### **Wird ein deutsches BERT Modell für klinische Sprache genutzt?**

Nein. Es existieren aktuell nur vereinzelt deutsche BERT-Modelle z.B. GottBERT (18) oder GermanBERT (19). Weder die genannten BERT-Modelle noch weitere deutsche BERT-Modelle wurden mit klinischen Daten trainiert, somit ist ein Einsatz für diese Bachelorarbeit nicht sinnvoll. Stattdessen werden die Eingangstexte aus den Dokumenten von SIWF automatisiert ins Englische übersetzt. Dies kann theoretisch zu Fehlern in der Übersetzung führen. Da jedoch BERT über ein immenses englisches Vokabular verfügt und zudem den Kontext berücksichtigt, gehen wir davon aus, dass dieser eine eher untergeordnete Rolle spielt.

### **Warum wird kein bestehendes BERT-Modell innerhalb dieser Bachelorarbeit auf die deutsche klinische Sprache trainiert und verwendet?**

Für das hierfür erforderliche Pretraining eines deutsch-klinischen BERT-Modells braucht es folgende Voraussetzungen:

#### Ausreichend viele und variable Trainingsdaten

Die angestrebten Datenbasis aus 83 Weiterbildungsprogrammen des SIWF und den 800 Leitlinien von der Arbeitsgemeinschaft der Wissenschaftlichen Medizinischen Fachgesellschaften (AMWF) würde für das Pretraining sehr wahrscheinlich nicht ausreichen. Es müssten zusätzlich weitere deutschsprachige, medizinische Ressourcen gefunden werden um ein robustes, deutsches, klinisches BERT-Modell zu trainieren.

#### Ressourcen (leistungsstarke Hardware und Zeit)

In einer wissenschaftlichen Arbeit, in welcher ein BERT-Modell auf klinischen Daten aus 1. Mio. elektronischen Krankenakten trainiert wurde, dauerte das Pretraining über zwei Wochen mit einem Tesla P40 GPU (9). Das Pretraining von ClinicalBERT dauerte 18 Tage mit einem GeForce GTX TITAN X 12 GB GPU (20). Innerhalb der Bachelorarbeit fehlt uns die Zeit und die notwendige Hardware für ein Pretraining. Zudem liegt der Fokus der Bachelorarbeit auf einer unterstützenden Integration des Prototyps im Geschäftsprozess des SIWF und nicht auf der Entwicklung eines neuen deutschen BERT-Modells.

Aus den obengenannten Gründen wird auf ein Pretraining in dieser Arbeit verzichtet.

### **Werden (Betriebs-) Kosten der Entwicklung und Applikation betrachtet?**

Nein, auf eine Kosten- und Leistungsrechnung wird verzichtet. Der Fokus der Thesis liegt auf der Mach- und Umsetzbarkeit einer NLP-Anwendung im Rahmen eines industriellen Anwendungsfalls. Das Ziel ist ein Prototyp und keine marktreife Software. Bei einer Weiterführung/-entwicklung des Projektes muss dieser Bereich noch genauer betrachtet werden.

### **Werden Massnahmen in Bezug auf den Datenschutz ergriffen?**

Nein, in dieser Arbeit werden keine «besonders schützenswerte Personendaten» verwendet.

## 2 Grundlagen

In diesem Kapitel werden die dieser Arbeit zu Grunde liegenden Technologien und Konzepte erläutert. Es werden u.a. der Begriff Natural Language Processing sowie die Architektur von Transformers und das BERT-Modell erläutert.

### 2.1 Natural Language Processing

Natural Language Processing ist ein Anwendungsfeld der Informatik und Linguistik. Es ist im Bereich der künstlichen Intelligenz anzusiedeln. NLP wird heute in verschiedensten Anwendungsgebieten eingesetzt, u.a. für Chatbots, Übersetzungsdienste wie DeepL und in digitalen Assistenten wie Siri oder Alexa. Einer der etabliertesten Anwendungsfällen von NLP ist die Textklassifizierung (engl. Text Classification). Etwa der Spam Filter in den Mäildiensten (Gmail, Outlook etc.), bei welchen die eingehenden E-Mails anhand des Betreffs dahingehend klassifiziert werden, ob es sich um eine Spam-Mail handelt oder nicht.

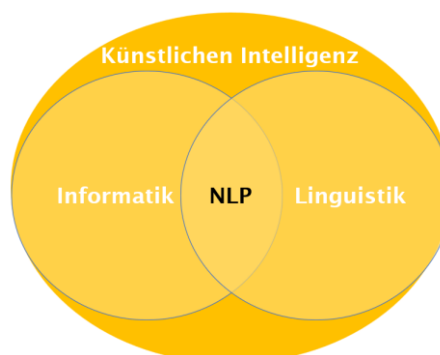


Abbildung 2: Anwendungsfeld NLP

Im Prinzip werden beim NLP verschiedene Techniken und Methoden des maschinellen Lernens (Machine Learning) und Deep Learning genutzt, um natürliche Sprache (gesprochen und geschrieben) zu erfassen und mit Hilfe von Algorithmen maschinell verarbeiten zu können.

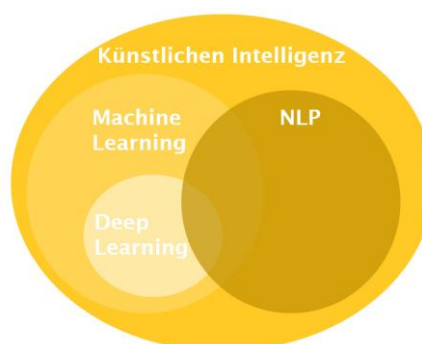


Abbildung 3: Bereich NLP in KI

Natural Language Processing besteht im Kern aus demselben Ablauf wie die Kommunikation zwischen Menschen. Es wird eine Information des Senders via ein Medium codiert (Input), vom Empfänger decodiert (Verarbeitung) und es erfolgt eine Rückmeldung durch eine Antwort oder indem die erhaltene Information als Wissen gespeichert wird (Output). Damit Maschinen diese Aufgabe ähnlich gut wie der Mensch lösen können, hat sich die Informatik u.a. an der biologischen Struktur des menschlichen Gehirns orientiert. Aus diesem Ansatz wurde das Modell der künstlichen Neuronalen Netze (kNN) entwickelt (Kapitel 2.1.3).

### 2.1.1 Word Embeddings

Sprache ist vielschichtig und oft auch mehrdeutig. Das semiotische Dreieck stellt die Komplexität von Sprache vereinfacht dar:

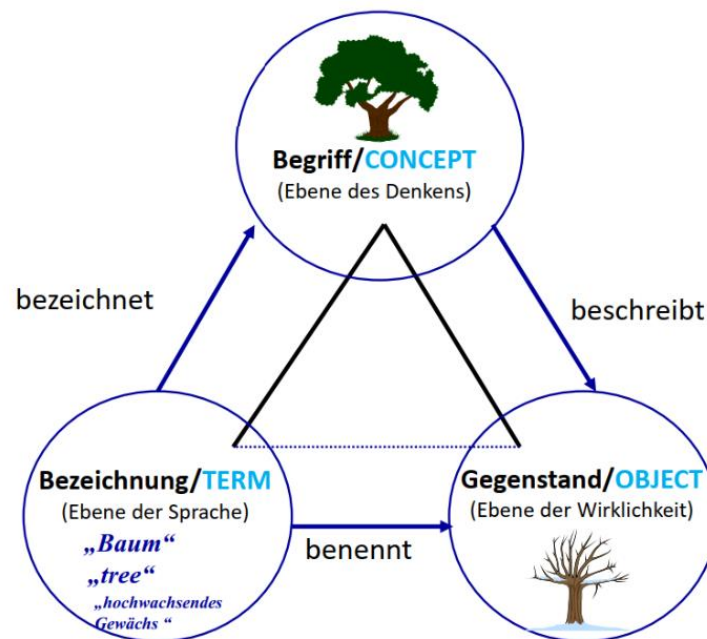


Abbildung 4: Semiotisches Dreieck (21)

In der Abbildung 4 wird die Sprache als Bezeichnung dargestellt. Die Bezeichnung besteht aus Symbolen, meist hör-, sichtbar oder anders wahrnehmbar. Diese Bezeichnung benennen Gegenstände/Sachverhalte/Ereignisse. Diese Benennung geschieht jedoch nur indirekt. Jeder Gegenstand wird auf der Ebene des Denkens durch ein Konzept abgebildet, welches den Gegenstand beschreibt. Die Bezeichnung bildet das Konzept ab und somit nur indirekt den Gegenstand.

Problematisch wird es, wenn der Gegenstand, das Konzept und die Bezeichnung nicht eindeutig zusammengeführt werden können. So entsteht Mehrdeutigkeit.

Die Auflösung der Mehrdeutigkeit (Ambiguität) einer Bezeichnung kann nur über den Kontext der Bezeichnung geschehen. Klassisches Beispiel hierfür ist das Wort "Bank". Dies kann sowohl ein Geldinstitut als auch eine Sitzgelegenheit repräsentieren. Die Bezeichnung alleine lässt nicht darauf schließen, um welchen Gegenstand es sich nun handelt. Für eine saubere Auflösung werden mehr Informationen über das Konzept der Bezeichnung benötigt (Kontext), damit auf den richtigen Gegenstand verwiesen werden kann. Diese Komplexität stellt die maschinelle Verarbeitung von Sprache vor grosse Herausforderungen.

Der in der natürlichen Spracherkennung verwendete Ansatz sind Word Embeddings (dt. Worteinbettung). Vereinfacht erklärt sind Word Embeddings die Zahlenrepräsentation von Wörtern. Bei einer Worteinbettung handelt es sich um eine Einbettung im mathematischen Sinne. Dies bedeutet, dass es sich um eine Abbildung handelt, die es ermöglicht, ein Objekt als Teil eines anderen zu erfassen. Diese Erfassung wird durch einen Vektor mit n-reellen Zahlen erreicht. Das Ziel des Vektors ist, eine abstrakte Darstellung der Bedeutung der Bezeichnung bei gleichzeitiger Dimensionsreduktion zu erhalten. Dabei bezieht sich der Vektor immer nur auf den verwendeten Korpus, aus welchem das Vokabular entstanden ist. Durch die Repräsentationen der Worte werden diese maschinell verarbeitbar und Zusammenhänge oder Beziehungen von Worten werden berechenbar.

Neuere NLP-Modelle, wie beispielsweise BERT, haben nicht nur einen statischen Wortstamm, sondern auch einen kontextabhängigen. Dies wird durch eine mehrdimensionale Repräsentation eines Wortes und der Berechnung des Kontextes erreicht.

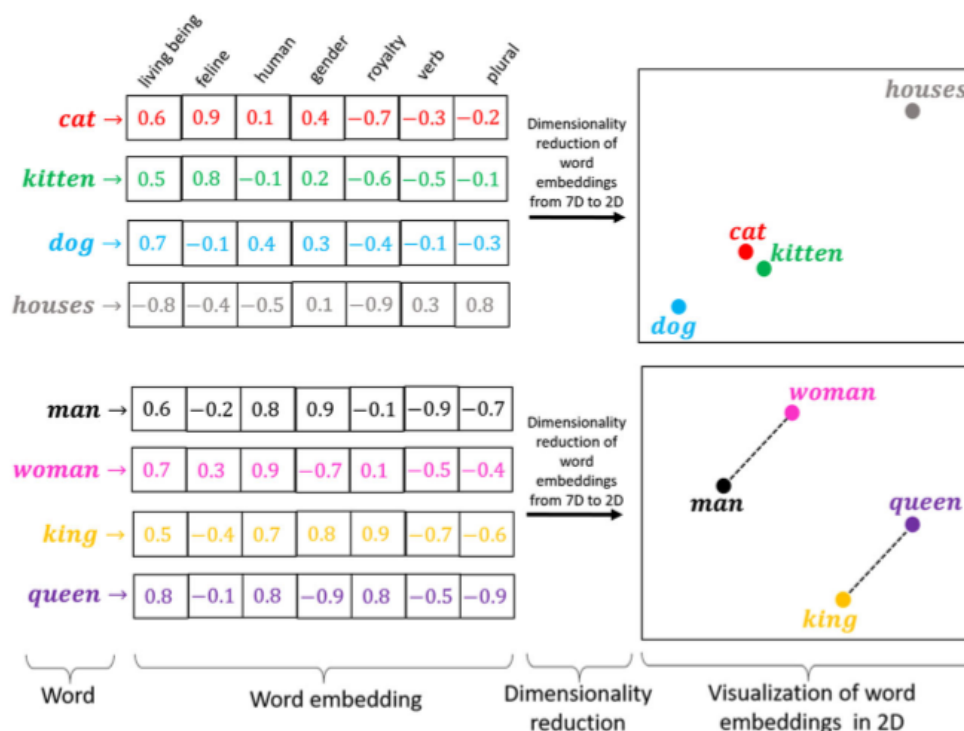


Abbildung 5: Schematische Darstellung von Word Embeddings (22)

Die oben aufgeführte Darstellung zeigt das Konzept des Word Embeddings von links nach rechts. Ein Wort wird anhand einer beliebigen Anzahl Eigenschaften numerisch zwischen -1 und 1 bewertet. Beispielsweise haben die Wörter Mann, Frau, König und Königin bei der Eigenschaft «lebendiges Wesen» (engl. living being) und Mensch (engl. human) einen Wert nahe bei 1, d.h. das Wort erfüllt diese Eigenschaft. Hingegen haben dieselben Wörter korrekterweise bei der Eigenschaft katzenartig (engl. feline) einen tiefen Wert. Durch diese Vektorisierung entlang von Eigenschaften ist eine Maschine beispielsweise in der Lage folgende Berechnung durchzuführen:  $\text{King} - \text{Man} + \text{Woman} = \text{Queen}$ . Die Addition der Word Embeddings der Wörter König und Frau und die Subtraktion von Mann ergibt das Word Embedding für Königin. Dieses Beispiel beschreibt, wie Word Embeddings die Beziehung des Geschlechts erfassen können (siehe Abbildung 5, schwarz gestrichelte Linie, beispielsweise zwischen Mann und Frau).

## 2.1.2 NLP-Modell

Die Kernaufgabe eines NLP-Modells besteht darin, den Input (geschrieben oder gesprochene Sprache) maschinell verarbeitbar zu machen und einen bestimmten Aufgabenbereich zu erfüllen. Beispielsweise Autokorrektur oder Suchvorschläge errechnen. In unserem Fall das Erkennen des Inhaltes im Text und eine entsprechende Klassifizierung.

Bei der Erstellung eines NLP Modells wird jedes Wort oder Wortteil (engl. Token) aus einem Textkorpus vom Modell aufgenommen, in Relation zu allen anderen bereits existierenden Wörtern gesetzt und erhält so seine Position im Vokabular des Modells. Diese Position wird durch die Worteinbettung repräsentiert und macht das Wort für das Modell für nachfolgende Aufgaben berechenbar.

In der vorhergegangenen Living Case 2 Arbeit hat das ausgewählte NLP-Modell die Bedeutung der einzelnen Wörter nur über statische Worteinbettungen (engl. Static Word Embedding) erkannt. Bei der Verarbeitung von Static Word Embedding wird der Kontext eines Wortes nur vom aktuell betrachteten Wort aus erfasst. Also nur ein Fenster von  $n$  Wörtern vor und hinter dem aktuell betrachteten Wort, nie der gesamte Kontext eines Satzes bzw. Inputs.

Beim verwendeten Tool der Living Case 2 Arbeit wurde das NLP-Modell genutzt, um Wörter bzw. Wortabfolgen in einem Text zu finden, welche mit einem in einer "Label-Datenbank" vorhandenen SNOMED CT Konzepte übereinstimmten.

Nüchtern betrachtet wurde somit nur ein Abgleich von gleichen Repräsentationen gemacht. Zudem schloss dieser Ansatz eine mehrfache Zuteilung einer Zeichenfolge zu Konzepten aus. Dieser Ansatz führte die Living Case 2 Arbeit zu ernüchternden Ergebnissen.

Mit BERT als neues NLP-Modell für unseren Anwendungsfall werden durch kontextabhängige Wortembeddings (engl. Contextual Word Embedding) die Wörter und dadurch der Kontext besser erkannt. Zudem wird für die Klassifizierung der Gesamtkontext des Inputs verwendet und gleichzeitig die Wahrscheinlichkeit für eine Zugehörigkeit zu allen SNOMED CT Konzepten geprüft.

Nachfolgend stark vereinfachte Abbildung 6 soll den groben Ablauf unseres NLP Modells darstellen.

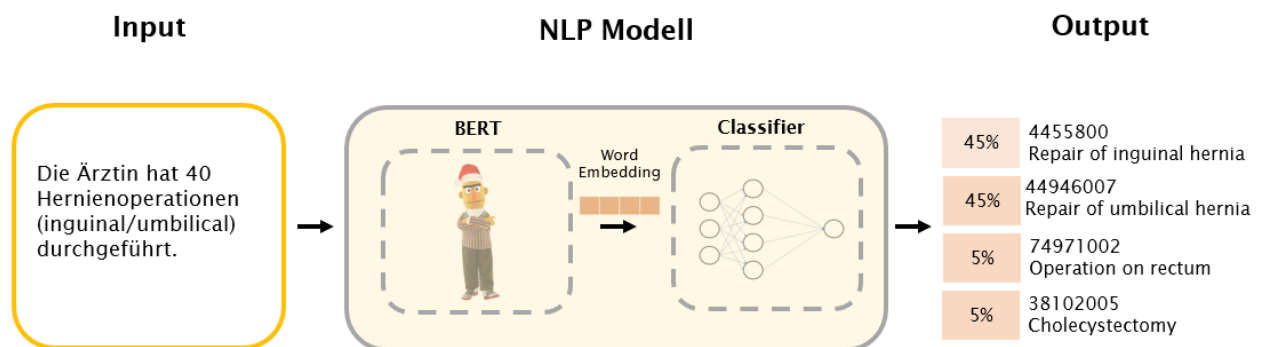


Abbildung 6: Ablauf NLP mit NLP Modell und Classifier

### 2.1.3 Neuronale Netze

Ein neuronales Netz besteht aus  $n$  Neuronen, die miteinander über mehreren Schichten (engl. Layer) verknüpft sind. Bei jedem neuronalen Netz gibt es einen Inputlayer, Hiddenlayer (versteckte Schicht) und Outputlayer.

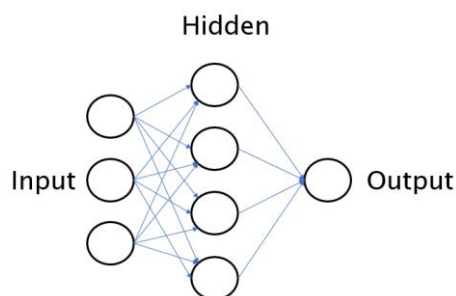


Abbildung 7: Architektur neuronales Netz

Innerhalb des Hidden Layers wird der Input in den gewünschten Output überführt. Die Neuronen sind durch Kanten miteinander verbunden. Jedes Neuron ( $X_i$ ) führt die Inputs zusammen und reagiert seiner Aktivierungsfunktion entsprechend darauf. Hierfür werden die Inputs gewichtet ( $W_i$ ) und einer Aktivierungsfunktion ( $f(x)$ ) übergeben. Der Output der Aktivierungsfunktion ( $Y$ ) ist der Input für die Neuronen im nächsten Layer. Wie und ob ein Neuron aktiviert wird, hängt von der Summe der Aktivierungsfunktion und dem sogenannten «Bias» ab. Die Aktivierungsfunktion berechnet, aus der Summe der erhaltenen Gewichte, ob sie ein Signal an die Neuronen der nächsten Schicht weitergibt oder nicht. Mit Hilfe des Bias kann der Output beeinflusst werden, so dass z.B. bei einem negativen Summenergebnis das Neuron trotzdem aktiviert wird. Abbildung 8 zeigt das Grundprinzip der Informationsweiterleitung eines neuronalen Netzes, genauer gesagt eines Feed-Forward Netzes.



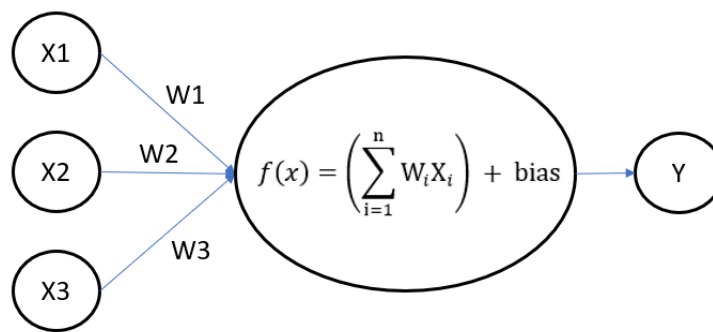


Abbildung 8: Aktivierungsfunktion neuronales Netz (Feed-Forward-Netz)

Ein Modell, dessen Outputlayer auf den oben aufgeführten Grundprinzipien eines Feed-Forward neuronalen Netzes basiert, kann nun mittels überwachtem Lernens (engl. Supervised Learning) für einen spezifischen Anwendungsfall (bspw. Klassifizierung) trainiert werden. Für eine Klassifizierungsaufgabe wird eine solche Schicht als Klassifizierer (engl. Classifier) bezeichnet. Voraussetzung hierfür ist jedoch, dass ein bereits ausreichend vortrainiertes (engl. pretrained) Modell – respektive vortrainierte Gewichte – vorhanden sind.

Für unseren Anwendungsfall wird ein NLP-Modell von Google (BERT) verwendet. Wie bereits im Kapitel 1.5 aufgeführt, wird auf eine Initialerstellung eines eigenen BERT-Modells verzichtet und stattdessen auf ein vortrainiertes BERT-Modell von Google gesetzt. Das Übernehmen der Gewichte eines bestehenden Modells wird u.a. als Transfer Learning bezeichnet.

## 2.2 Transfer Learning

Transfer Learning eines neuronalen Netzes bedeutet, ein bestehendes Modell auf eine spezifische Domäne oder Aufgabe hin anzupassen. Somit wird die Struktur des Modells mit den bestehenden Layern und gelernten Gewichten übernommen und als Ausgangspunkt für die Entwicklung eines neuen Modells verwendet. Für die Veränderung von bestehenden Modellen gibt es verschiedene Möglichkeiten oder Strategien. Dazu gehören das Pretraining und das Finetuning eines Modells. Folgend werden die Bereiche Pretraining und Finetuning etwas genauer beleuchtet.

### 2.2.1 Pretraining

Das Pretraining eines neuronalen Netzes zielt darauf ab, ein bestehendes Modell auf eine spezifische Domäne generisch anzupassen. Für das Pretraining eines neuronalen Netzes wird ein grosser Datensatz benötigt, welcher für die gewünschte Domäne repräsentativ ist. Mittels unüberwachtem Lernen (engl. unsupervised learning) lernt das neuronale Netz selbstständig aus dem ihm vorgelegten Datensatz und verändert die Gewichte des gesamten Modells.

Im Rahmen von NLP wird bei diesem Prozess zusätzlich auch das Vokabular erweitert und die Repräsentationen der einzelnen Wörter (engl. Word Embeddings) im Gesamtkontext des Vokabulars (Korpus) angepasst. Somit wird das Modell generisch auf einen anderen Verwendungsbereich erweitert. Das ursprüngliche BERT-Modell ist auf Basis von englischen online Büchern und Wikipedia-Einträgen trainiert worden. Somit ist dessen Domäne sehr allgemein und relativ unspezifisch gehalten. Durch Pretraining mit englischen PubMed Abstracts und Volltext Journals aus PMC wurde das BERT-Modell spezifisch für die medizinischen und wissenschaftlichen Domäne entwickelt und unter dem Namen BioBERT veröffentlicht (11).

### 2.2.2 Finetuning

Das Finetuning eines neuronalen Netzes zielt darauf ab, ein bestehendes Modell auf eine spezifische Aufgabe zu trainieren. Für das Finetuning eines neuronalen Netzes benötigt es – im Gegensatz zum Pretraining – einen gelabelten Datensatz, welcher für die gewünschte Aufgabe repräsentativ ist. Dieser Datensatz beinhaltet die Daten, welche dem Modell als Input gegeben werden und einer Repräsentation vom jeweiligen erwarteten Output. Durch diesen Datensatz kann das neuronale Netz nun überwacht trainiert (engl. supervised learning) werden. Somit lernt das Modell anhand von Beispielen, was erwartet



wird und verändert entsprechend die interne Struktur (Gewichtungen) der verborgenen Schichten (engl. Hidden layer).

Beim Finetuning wird vor dem Training entschieden, ob und an welchen verborgenen Schichten Anpassungen vorgenommen werden sollen. Zudem können dem ursprünglichen Modell auch zusätzliche Schichten mit Neuronen zugefügt werden, welche einender Aufgabe entsprechenden, Output liefern. Beispielsweise werden bei einer Textklassifizierung, einem BERT-Modell 1 - n Schichten als Classifier angehängt. Einerseits muss nun das Modell lernen, welcher Input zu welcher Output-Klasse führt und zusätzlich auch wie der gewünschte Output aussehen soll.

Im Kontext dieser Arbeit ist der Input eine Abfolge von Wörtern. Deren Kontext, am Ende der Verarbeitung im BERT-Modell, durch den Classifier verarbeitet wird. Der Output des Classifier ist eine prozentuale Angabe bezüglich Übereinstimmung auf jedes dem Modell bekannten SNOMED CT Konzeptes. Für das Training des Classifier wird ein gelabelter Datensatz benötigt, welcher Beispielsätze oder Textabschnitte und eine jeweilige Zuweisung zu dem im Text enthaltenen SNOMED CT Konzept enthält. Mehr Details siehe Kapitel 3.4.2.2 Bereich Finetuning.

## 2.3 BERT, Transformers und Attention

Bidirectional Encoder Representations from Transformers, oder kurz BERT, basiert auf dem Konzept der Transformers, einer Deep Learning Architektur mit «Attention», die ebenfalls von Google 2017 veröffentlicht wurde (23). Abbildung 9 zeigt ein Transformer-Modell. Es besteht aus zwei Hauptkomponenten, den Encoders und Decoders. Jeder Encoder selbst wiederum besteht aus einem Self-Attention Modul und einem neuronalen Netz. Jeder Decoder beinhaltet ein neuronales Netz, ein Self-Attention- sowie Encoder-Decoder Attention Modul.

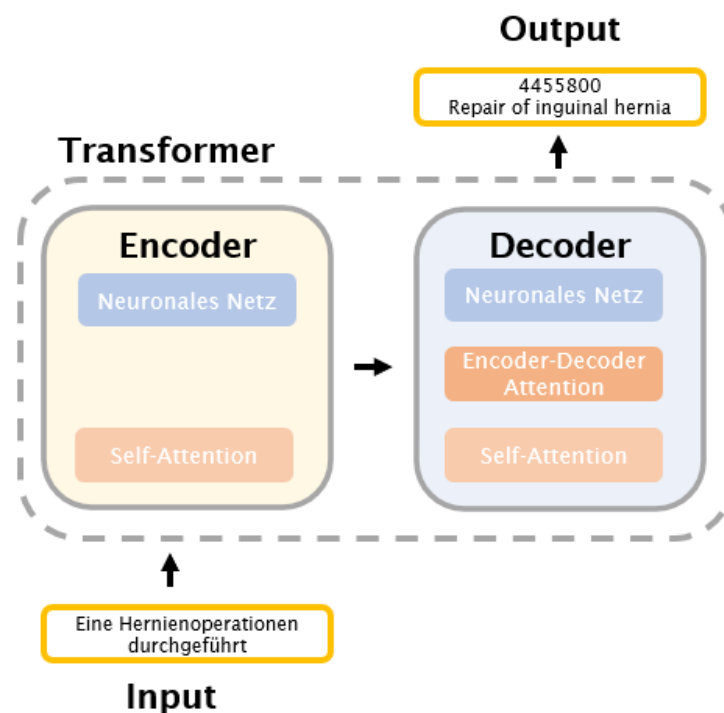


Abbildung 9: Transformer-Modell mit Encoder und Decoder

Während das Transformer-Modell jeden Token verarbeitet, kann es durch Anwendung des Self-Attention Layers bei anderen Tokens im Inputtext nach Hinweisen suchen, die zu einer besseren Repräsentierung für das gerade betrachtete Wort führen. Die Self-Attention ist somit das Gedächtnis des Transformer-Modells, angewendet auf die gerade betrachtete Inputsequenz.

Diese ermöglicht es dem Modell, die anderen Wörter in der Eingabesequenz zu betrachten, um ein bestimmtes Wort in der Sequenz besser zu verstehen. Somit kann das Modell zum Beispiel lernen, dass das Wort «durchgeführt» mit den Wörtern «Hernien Operationen» und «eine» assoziiert ist. Nachfolgende Abbildung 10 zeigt die Self-Attention visualisiert am Beispielsatz: «The animal didn't cross the street because it was too wide». Wenn das Modell links das Wort "it" verarbeitet, wird die Self-Attention die Wörter, "it", "animal» und «street» im selben Satz assoziieren, wobei die höchste Attention korrekterweise dem Wort «street» attestiert wird.

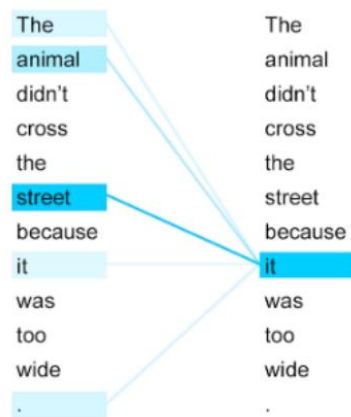


Abbildung 10: Self-Attention Beispiel (24)

Das BERT Modell besteht nun aus mehreren hintereinander gekoppelten Encodern ( $BERT_{small} = 12$  Encoders,  $BERT_{large} = 24$ ), siehe Abbildung 11. Beim ersten Encoder wird jeder Token vom Inputtext in einen Vektor der Grösse 768 transferiert, zudem wird beim ersten Word Embedding, die Position des Tokens in der Inputsequenz integriert und ein Spezialtoken (CLS für Classification) mitgeliefert [1]. Die Vektoren durchlaufen anschliessend die Self-Attention [2], dann das neuronale Netz [3]. Die neuen Vektoren werden dann dem nächsten Encoder übergeben [4], bei welchen die vorherigen Subschritte wiederholt und dem nächsten Encoder weitergereicht werden usw.

Der CLS Token beinhaltet am Ende alle Informationen der Inputsequenz, oder in anderen Worten: Der CLS Token repräsentiert den Kontext des Satzes und wird daher für Klassifizierungsaufgaben als Eingabe für den Classifier verwendet [5].

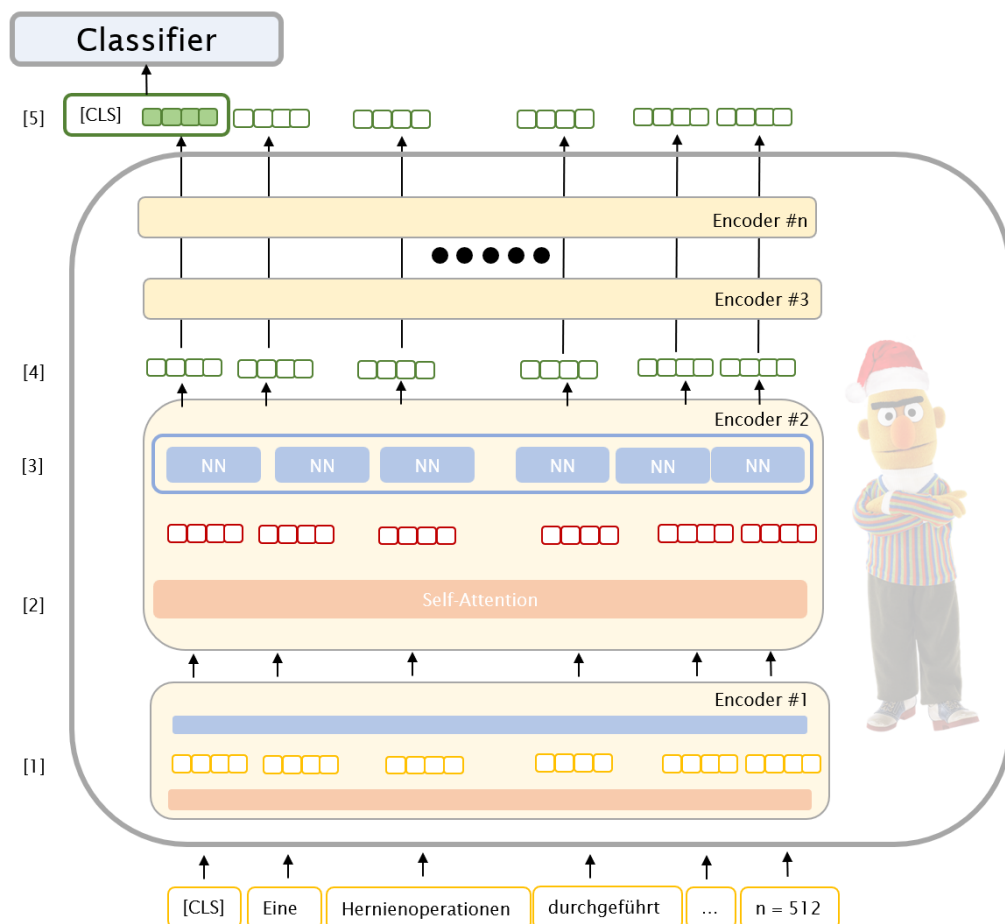


Abbildung 11: Encoder Stack mit Self-Attention und neuronalen Netz

Dass der CLS-Token für den Classifier genommen wird, liegt daran, dass das BERT-Modell mit den zwei Hauptaufgaben «Masked Language Model» und «Two-Sentence Classification» trainiert wurde. Bei der maskierten Sprachmodellierung werden im Inputsatz zufällige Wörter mit einem [MASK]-Token maskiert. Das BERT-Modell lernt innerhalb des Pretrainings diese Wörter vorherzusagen.

Bei der Two-Sentence Classification lernt das BERT-Modell anhand zweier Inputsätze vorherzusagen, ob der zweite Satz der richtige Satz ist, der auf den ersten Satz folgt. Für diese Aufgabe wird der CLS-Token verwendet. Er gibt an, wie wahrscheinlich der aktuelle Satz, der auf den ersten Satze folgende ist. Wenn es um Klassifizierungsaufgaben geht (bspw. die Vorhersage des passenden SNOMED CT Konzepts für eine Prozedur), dann ist der CLS-Token sehr hilfreich, da dieser die Repräsentation auf Satzebene enthält. Diese Repräsentation auf Satzebene kann auch als Kontext des Satzes verstanden werden und wird als eigenständiges Word Embedding dargestellt. Dieses Word Embedding platziert sich nun im Vektorraum des gesamten Vokabulars was als Ausgangspunkt für weitere Berechnungen verwendet werden kann.

## 2.4 SNOMED CT

SNOMED CT entstand aus einer Kollaboration des Colleges of American Pathologists und dem National Health Service. Der Gedanke dahinter ist, eine Datenbank mit Begriffen zu erstellen, die es ermöglicht die ganze Medizin in Form von Konzepten zu erfassen und somit abbilden zu können. Die ontologiebasierte Nomenklatur SNOMED CT hat das Potential, die semantische Interoperabilität zu erleichtern (25). Das SIWF hat sich für diese internationale Nomenklatur als technischen Grundlagen für einen einheitlichen Lernzielkatalog (definiert in den Weiterbildungsprogrammen) entschieden. Deswegen wird in dieser Arbeit ein Mapping auf SNOMED CT Konzepte vorgenommen.

## 3 Methodik

Die Vorgehensweise um die Fragestellungen beantworten und die Ziele erreichen zu können, wird in den nachfolgenden Unterkapiteln vorgestellt. Zu Beginn wird auf die Projektorganisation mit den wichtigsten Punkten, wie Meilensteine, Risiken und Planung eingegangen. Im zweiten Abschnitt steigen wir in die Methodiken der Informationsbeschaffungen, wie Anforderungsanalyse und Recherche, ein. Am Schluss dieses Kapitels wird der für diese BSc. Thesis bedeutendster Teil – die Vorgehensweise für die Konzepterarbeitung und Erreichung der beiden Hauptziele – erläutert. Da sich der wissenschaftliche Zweck dieser Arbeit auf den letzten Abschnitt bezieht, werden in den vorhergehenden Abschnitten (Projektmanagement, Anforderungsanalyse und Recherche) direkt Ergebnisse präsentiert.

### 3.1 Projektmanagement

Wir haben diese BSc. Thesis mittels hybrider Projektmanagement-Methoden verwaltet. Die Arbeit wurde auf drei Phasen mit neun Arbeitspaketen und vier Meilensteinen verteilt. Die Phasen waren: Initialisierung, Konzept und Realisierung. Die Initialisierungs- und Konzeptphase ist nach der klassischen Projektmanagement Methodik (Plan Driven) erarbeitet worden. Innerhalb der Realisierungsphase wurde die agile Vorgehensweise nach Scrum angewendet. Details zu den Arbeitspaketen sind im Anhang 10.2 abgelegt. Die Meilensteine werden in nachfolgender Tabelle erläutert. Weitere Informationen zu den Meilensteinen sind im Anhang 10.3 zu finden.

ID	Meilenstein	Beschreibung
M1	Anforderungs-analyse	In der Anforderungsanalyse wurde die Aufgabenstellung mit den Dozierenden und dem SIWF geklärt. Die Erwartung des Experten abgeholt und die Deliverables festgelegt. Der Meilenstein ist erreicht, wenn das Pflichtenheft durch den Betreuer und vom SIWF abgenommen worden ist.
M2	Detaillkonzept	Im Detailkonzept wurde das Grobkonzept weiter vertieft und konkretisiert. Das Detailkonzept beinhaltet die finale Systemarchitektur, das Vorgehen für das Self-supervised Learning und dem Weg wie das Endprodukt ein Entity-Linking mit F1 Werte >70 % erzielen soll. Der Meilenstein ist erreicht, wenn der Betreuer «grünes Licht» für die Weiterentwicklung des Detailkonzepts erteilt hat.
M3	Prototyp	Der Prototyp wurde nach der agilen Entwicklungsmethode Scrum in 5 Sprints à 2 Wochen umgesetzt. Der Code wurde in ausreichender Form dokumentiert. Der Meilenstein ist erreicht, wenn der Prototyp mit den BERT-Modellen umgesetzt, der SUS Score vom SIWF erhoben und am letzten Scrum Review Rückmeldungen abgenommen worden sind.
M4	Deliverables	Alle unter Deliverables aufgeführten Items sind vorhanden. Der Meilenstein ist erreicht, wenn alle Items zeitgerecht und an die definierten Stellen abgegeben worden sind.

Tabelle 2: Übersicht Meilensteine BSc. Thesis

#### 3.1.1 Organisation

Damit alle relevanten Informationen jederzeit den Projektpartnern und dem Betreuer zugänglich sind, wurde eine Projektmappe in Microsoft OneNote erstellt. Darin wurden alle administrativen Arbeiten (Protokolle, Kontaktliste, Pendenzenliste, Fragen- und Antworten Katalog etc.) verwaltet. Weiter wurden mit dem Betreuer eine Terminserie für jeden Mittwoch vereinbart, damit ein stetiger Austausch gewährleistet war.

### 3.1.2 Risikoanalyse

Wir führten zu Beginn der Arbeit eine Risikoanalyse nach Hermes durch (26). Es wurden sieben Risiken identifiziert. Ein hohes Risiko blieb, trotz zwei definierten und durch uns umgesetzten Massnahmen, bestehen.

ID	Typ	Beschreibung
R6	Hohes Risiko	Unterschätzung der Komplexität der Umsetzung der Systemarchitektur in einen Prototyp. Beispielsweise aufgrund mangelnd geführter und validierter Dokumentation der ausgewählten Systemkomponenten (Sprachmodell, Framework, Server).
M6.1	Massnahme	Detaillkonzept (Meilenstein Nr. M2) von SIWF und Betreuer (Erfahrung von Betreuer abholen) abnehmen lassen.
M6.2	Massnahme	Intensive Recherche (Literatur & Internet) zu ähnlichen Projekten. Arbeitspaket «Recherche» läuft parallel zu dem Arbeitspaketen «Grobkonzept», «Detaillkonzept» und anfangs «Prototyp».

Tabelle 3: Ergebnis und Massnamen aus Analyse hohes Risiko

Die komplette Risikoanalyse mit den weiter definierten Massnahmen, Verantwortlichkeiten und Fristen wird im Anhang 10.4 aufgeführt.

### 3.1.3 Planung

Der personelle Aufwand wurde auf 720 Stunden während 17 Wochen geschätzt und beträgt pro Person etwas über 20 Stunden die Woche. Für die Planung ist nachfolgendes Gantt-Diagramm (Abbildung 12) erstellt worden. Die Erläuterungen zu den einzelnen Arbeitspaketen werden im Anhang 10.2 aufgeführt.

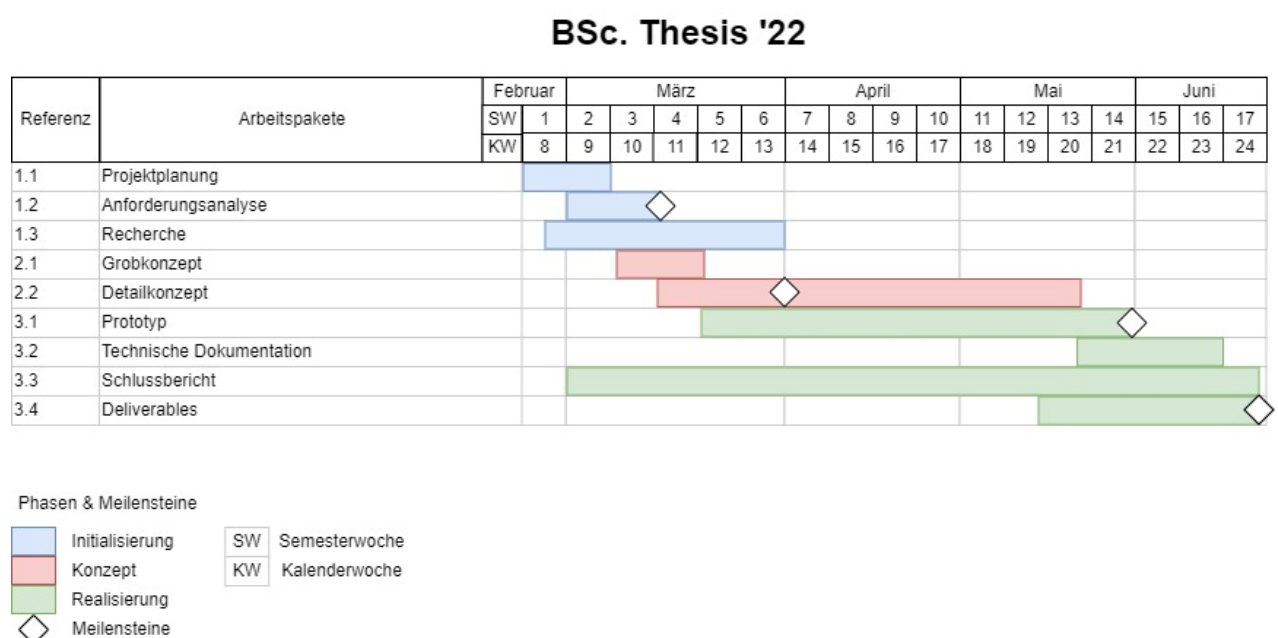


Abbildung 12: Gantt-Diagramm BSc. Thesis

### 3.2 Anforderungsanalyse

In der Anforderungsanalyse wurde die Aufgabenstellung mit den Dozierenden und dem SIWF geklärt. Die Erwartung des Experten abgeholt und die Deliverables festgelegt. Sie umfasste die Festlegung der am Ende der BSc. Thesis abzugebenden Lieferobjekte (engl. Deliverables), die Festlegung der Ziele der BSc. Thesis sowie die Dokumentation der funktionalen und nicht-funktionalen Anforderungen in einem Pflichtenheft (Anhang 10.5).

### 3.3 Recherche

Mit der Recherche haben wir uns weiter in die Themengebiete Natural Language Processing, Transfer Learning und BERT eingearbeitet und Ideen für die Konzeption der Software und des Modells gesammelt. Die Recherche umfasste eine Literatur- und Internetrecherche sowie eine allgemeine Informationsbeschaffung. Folgende Tabelle fasst die Recherche zusammen:

Typ	Beschreibung
Literatur-recherche	Es wurde mit 17 Suchwörtern in unterschiedlichen Kombinationen auf den Suchmaschinen Google Scholar und PubMed insgesamt 34 wissenschaftliche Arbeiten gefunden. Nach einer Analyse verblieben noch 20 relevante Arbeiten.
Internet-recherche	Innerhalb der Internetrecherche wurden über 15 Internetquellen gefunden.
Informations-beschaffung	Es sind 16 Weiterbildungsprogramme sowie die dazugehörigen Parametrierungskataloge (manuelle Annotation des Mappings zwischen textuellen Beschreibungen und SNOMED CT Konzepten) vom SIWF beschafft worden.

Tabelle 4: Zusammenfassung Recherche

### 3.4 Konzept

Um die beiden Hauptziele – Entwicklung Prototyp für Entity Linking und System Usability Score (SUS) (Kapitel 3.4.2.4) von mind. 68 Punkten zu erreichen – wird nebst den oben aufgeführten Methoden auch ein Detailkonzept auf Basis eines Grobkonzepts erarbeitet. Die einzelnen Arbeiten können grob in zwei Domänen unterteilt werden. Namentlich Arbeiten für die NLP Pipeline «NLP Pipeline» und Arbeiten für die Umsetzung der Desktopapplikation «Prototyp».

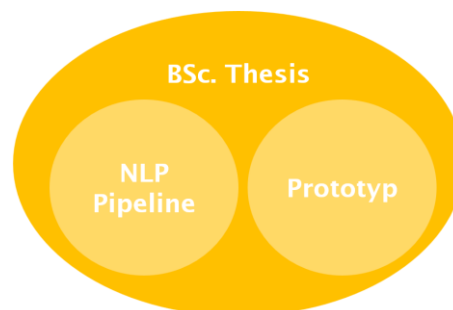


Abbildung 13: Domänen BSc. Thesis

In den nachfolgenden Unterkapiteln wird die Vorgehensweise für die Erarbeitung des Grobkonzepts sowie des darauf aufbauenden Detailkonzepts beschrieben. Weiter wird die Vorgehensweise für das Finetuning (Workflow Finetuning) und für die Gesamtevaluation beschrieben. Erarbeitete Ergebnisse wie Systemarchitektur, Ergebnis der Variantenanalyse sowie die Resultate aus der Gesamtevaluation der hier beschriebenen Vorgehensweise fließen in das finale Detailkonzept ein.

### 3.4.1 Grobkonzept

Der Zweck vom Grobkonzept ist es, die Grundlage für das Detailkonzept zu schaffen und die Vorgehensweise zu definieren, damit die, für diese Arbeit definierten Fragestellungen schlussendlich beantwortet werden können. Für das Grobkonzept sind u.a. folgende Arbeiten notwendig:

1. Abbildung des Geschäftsprozesses beim SIWF
2. Durchführung der Variantenanalysen
3. Erstellung der Systemarchitektur
4. Abbildung des Finetuning Workflows
5. Definition der Gütekriterien und Visualisierungsmöglichkeiten
6. Entwicklungsumgebung testen

Alle aufgeführten Arbeiten haben einen direkten oder indirekten Einfluss in den Technologieentscheid (Nr. 7). Zudem fließen die Ergebnisse der Arbeiten anschliessend in drei Bereiche des Detailkonzepts:

8. Finetuning
9. Umsetzung
10. Gesamtevaluation

Um darzustellen, welche Arbeiten in welchen Domänen stattfinden, wird die bereits aufgeführte Abbildung 13 mit den Arbeiten des Grobkonzepts und den drei Bereichen in der Abbildung 14 ergänzt.

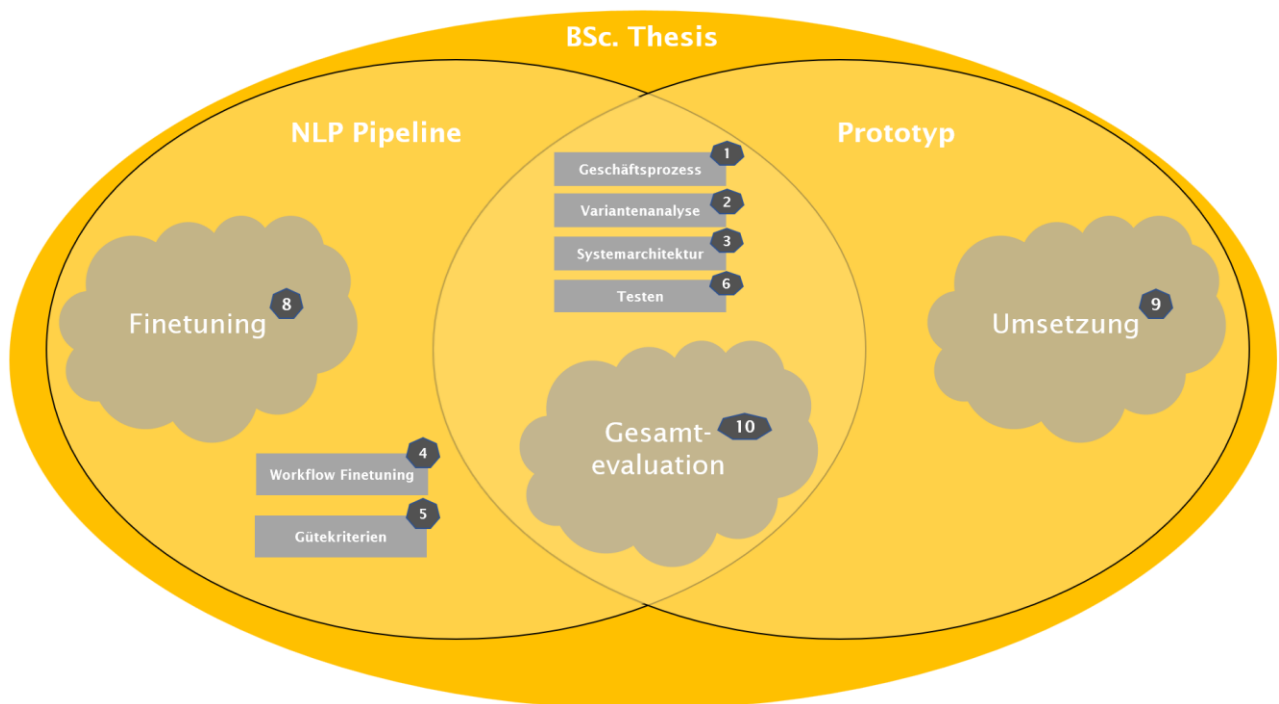


Abbildung 14: Detailansicht Domäne, Arbeiten und Bereiche



Um unsere Vorgehensweise für die Erarbeitung des Detailkonzepts weiter zu verdeutlichen, wird in nachfolgender Abbildung der Aufbau und die Struktur der oben aufgeführten Arbeiten abgebildet. Diese umfassen die drei Bereiche des Detailkonzepts und schlussendlich die Entwicklung des Prototyps, siehe Abbildung 15.

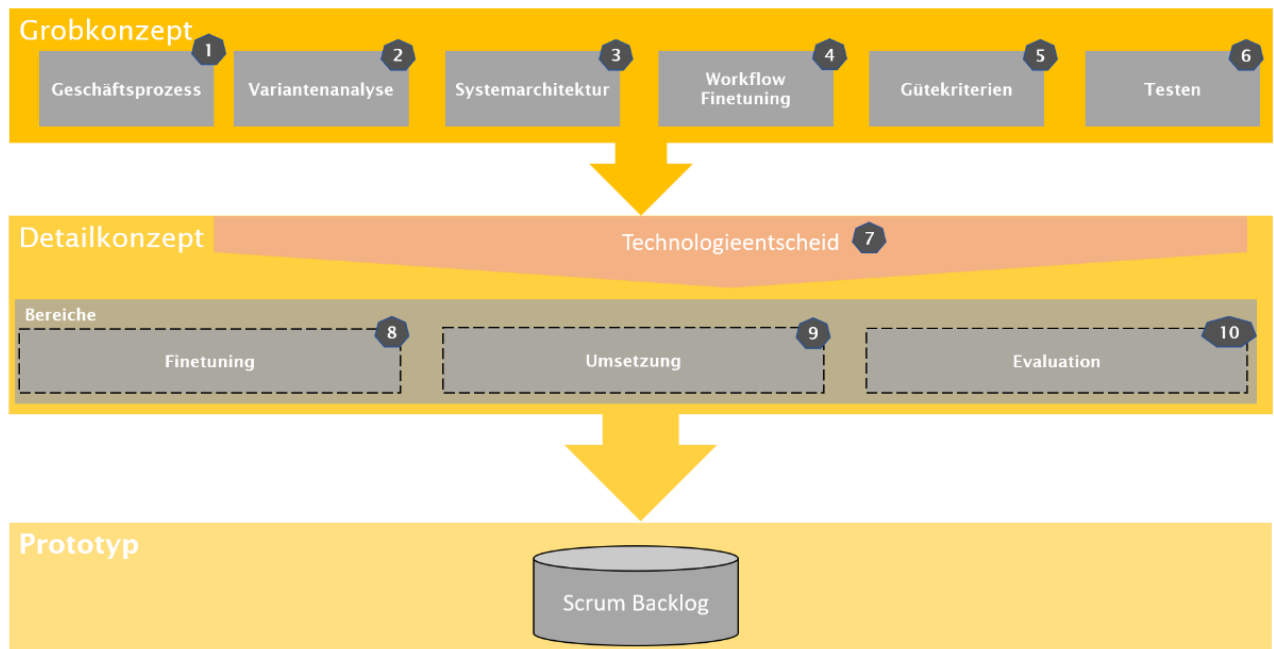


Abbildung 15: Ablaufdiagramm Konzept BSc. Thesis

Die einzelnen Komponenten sowie die übergeordneten Domänen (NLP Pipeline oder Prototyp) werden in untenstehender Tabelle weiter ausgeführt.

Nr.	Schritt	Beschreibung	Domäne
1	Abbildung des Geschäfts-prozesses beim SIWF	Der Prototyp soll das SIWF bei der manuellen Annotation des Mappings zwischen textuellen Beschreibungen und SNOMED CT unterstützen. Hierfür wird in Zusammenarbeit mit dem SIWF der aktuelle Geschäftsprozess mittels Business Process Model and Notation (BPMN) abgebildet.	NLP Pipeline / Prototyp
2	Durchführung der Variantenanalysen	Bevor ein Prototyp für den hier erforderlichen Anwendungsfall entwickelt wird, müssen wir uns für diverse geeignete Systemkomponenten entscheiden. Damit die Wahl der Komponenten (Technologieentscheid) nachvollziehbar ist, werden Variantenanalysen mit unterschiedlichen Bewertungskriterien durchgeführt.	NLP Pipeline / Prototyp
3	Erstellung der Systemarchitektur	Die Erstellung einer Systemarchitektur und dazugehörigem Datenfluss dient dem Zweck den Aufbau des Finetunings sowie des Prototyps und der damit zusammenhängenden Architektur zu verstehen.	NLP Pipeline / Prototyp

Nr.	Schritt	Beschreibung	Domäne
4	Abbildung des Workflows für das Finetuning des Modells	Im Rahmen dieser Arbeit werden zwei Classifier für ein BERT-Modell entwickelt. Der erste Classifier soll die Eingabe in zwei Bereiche «SNOMED CT Code» und «kein SNOMED CT Code» klassifizieren. Der zweite Classifier soll die Eingabe in die vorgegebene Anzahl SNOMED CT Konzepte klassifizieren und dient der Erreichung des Hauptziels Nr. 1. Für das notwendige Training sind diverse Vorarbeiten notwendig, die in einem dazugehörigen Workflow abgebildet werden.	NLP Pipeline
5	Definition der Gütekriterien und Visualisierungsmöglichkeiten	Um das zweite Hauptziel zu erreichen, werden wir Gütekriterien und deren Visualisierungsmöglichkeiten definieren. Durch eine einfache Visualisierung sollen die Gütekriterien Aufschluss geben können, ob die Word Embeddings für das SIWF Sinn ergeben und somit das BERT-Modell für eine weitere Verwendung nützlich sein wird.	NLP Pipeline
6	Entwicklungsumgebung testen	Die Umsetzung des Detailkonzeptes auf Codebasis geschieht in dieser Arbeit auf zwei unterschiedlichen Plattformen. Für das Finetuning wird Google Colab und für die Umsetzung der Desktopapplikation PyCharm verwendet. Google Colab bietet eine integrierte Versionskontrolle, welche von uns für die Zusammenarbeit verwendet wird. Für die Versionskontrolle des Python Projekts in PyCharm wird GitHub verwendet. Um eine möglichst reibungslose Zusammenarbeit gewährleisten zu können, wird das Setup der Entwicklungsumgebung vor der Umsetzungsphase mit den individuellen Computern getestet und die daraus gewonnen Erkenntnisse fließen direkt in die Umsetzung.	NLP Pipeline / Prototyp
7	Technologieentscheid	Fällung des Entscheids, welche Technologien für den Prototyp eingesetzt werden sollen und dem Entscheid, welches BERT-Modell verwendet werden soll.	NLP Pipeline / Prototyp
8	Bereich Finetuning	Erstellung der Systemarchitektur für die Entwicklung der BERT-Classifier sowie Darstellung des Ablaufs.	NLP Pipeline
9	Bereich Umsetzung	Erstellung der angestrebten Systemarchitektur für die Desktopapplikation.	Prototyp
10	Bereich Gesamtevaluation	Erarbeiten der Evaluationskriterien für die BERT-Modelle und für die Desktopapplikation.	NLP Pipeline / Prototyp

Tabelle 5: Kurzbeschreibung der Konzeptkomponenten

### 3.4.2 Detailkonzept

Alle durchgeführten Arbeiten innerhalb des Grobkonzepts fließen in die Erarbeitung der Bereiche Finetuning, Umsetzung und Gesamtevaluation ein. Die Resultate der Variantenanalyse haben einen direkten Einfluss auf den Technologie Entscheid (siehe Abbildung 15) und werden im Kapitel Ergebnisse präsentiert.

### 3.4.2.1 Technologieentscheid

Mittels Variantenanalysen wird entschieden, welche Technologien für den Prototyp eingesetzt werden, respektive welches BERT-Modell innerhalb der Domäne NLP Pipeline verwendet wird. Eine Übersicht der zu bewertenden Komponenten sowie der anzuwendenden Bewertungskriterien und eine Beschreibung der Vorgehensweise, wird in nachfolgender Tabelle aufgezeigt:

Komponente	Kriterien	Skala	Beschreibung
Domäne NLP Pipeline			
BERT-Modell	<ul style="list-style-type: none"><li>Verfügbarkeit</li></ul>	3 = Ja 0 = Nein	Auf Basis der Ergebnisse der Literaturrecherche und der Anforderungsanalyse werden die verfügbaren BERT-Modelle eruiert.
	<ul style="list-style-type: none"><li>State-of-the-art*</li></ul> <p>*Wie häufig wurde ein BERT-Modell in ähnlichen Forschungsfragen (Entity Linking auf SNOMED CT mittels Transfer Learning) verwendet.</p>	1 = gering 2 = mittel 3 = hoch	
Tool zur Visualisierung der Gütekriterien	<ul style="list-style-type: none"><li>Dokumentation</li><li>Flexibilität</li></ul>	1 = gering 2 = mittel 3 = hoch	Die Gütekriterien dienen nach dem Finetuning der Bewertung des entstandenen Modells. Durch eine einfache Visualisierung sollen die Gütekriterien Aufschluss geben können, ob die Word Embeddings Sinn ergeben und somit das Modell für eine weitere Verwendung nützlich sein könnte. Durch eine Internetrecherche, mit den Suchstrings «visualize» && «Word Embeddings», werden Visualisierungstools für die Gütekriterien gesucht. Die zu visualisierenden Gütekriterien sind: Principle Component Analysis (PCA) und t-Distributed Stochastic Neighbor Embedding (t-SNE) oder Uniform Manifold Approximation.
	<ul style="list-style-type: none"><li>Einarbeitungsaufwand</li></ul>	1 = hoch 2 = mittel 3 = gering	
	<ul style="list-style-type: none"><li>Interaktivität</li><li>Kompatibilität mit Colab</li></ul>	3 = Ja 0 = Nein	
Domäne Prototyp			
Server Framework	<ul style="list-style-type: none"><li>Dokumentation</li><li>Popularität</li></ul>	1 = gering 2 = mittel 3 = hoch	Im Pflichtenheft wird eine Funktionale „KANN“ Anforderung (ID: 8.FA) an den Prototyp definiert, mit dem Zweck, dass das Modell via REST API genutzt werden kann. Durch eine Internetrecherche, mit den Suchstrings «Python» && «REST API» && «Framework», werden mögliche Python Frameworks eruiert. Da es sich in dieser Arbeit um einen Prototyp handelt, werden für die Server Frameworks nur Micro Frameworks einbezogen.
	<ul style="list-style-type: none"><li>Einarbeitungsaufwand</li><li>Installationsaufwand</li></ul>	1 = hoch 2 = mittel 3 = gering	

Tabelle 6: Übersicht Methodik Variantenanalysen

Bei den in der Variantenanalyse aufgeführten Gütekriterien wird zusätzlich noch eine weitere Analyse einbezogen, welche aber erst dann zur Anwendung kommt, wenn Fehler bei der Vorhersage von SNOMED CT Konzepte auftreten. Hierfür wird die Cos-Similarity gewählt. Diese Analyse dient dem Ähnlichkeits-Vergleich von zwei Wörtern oder Sätzen im Rahmen des gegebenen Vektorraumes.

### 3.4.2.2 Bereich Finetuning

Wenn wir im Bereich Finetuning sind, befinden wir uns innerhalb der Domäne NLP Pipeline. Im Bereich Finetuning wird die Systemarchitektur für die Entwicklung der BERT-Classifizier konzipiert sowie der dazugehörige Workflow erstellt. Eine Dokumentation des geplanten Vorgehens im Rahmen des Modell-Training wird beschrieben.

Für die Erreichung des Hauptziels Nr. 2 benötigen wir zwei Classifizier. Der erste Classifizier soll eine Aussage machen, wie wahrscheinlich es sich beim Input um ein SNOMED CT Konzept vom Typ «procedure» oder «body structure» handelt oder nicht. Der zweite Classifizier soll eine Aussage machen, wie wahrscheinlich ein SNOMED CT Code – der vorhin genannten Typen – zu einem Input passt. Um bei den vorhin genannten zwei Classifiern das geeignete Supervised Learning machen zu können, braucht es mehrere Datensätze. Das bedeutet, dass wir für das Finetuning der Classifizier einen gelabelten Datensatz benötigen. Das heisst eine Liste mit Texten von klinischen Informationen und einer Kennzeichnung («SNOMED CT» oder «Not SNOMED CT») für jeden Eintrag und einen erweiterten Datensatz, der dieselbe Information erweitert mit dem SNOMED CT Code und Bezeichnung enthält. Die Herausforderung besteht darin, die gelabelten Datensätze für das Supervised-Learning der Classifizier zu erstellen. Grundsätzlich sind für das Finetuning zwei Teilaufgaben notwendig:

- Datenaufbereitung für das Supervised Learning (Data collection)
- Durchführung des Trainings (Supervised Learning)

#### Data collection

Wie bereits erwähnt, werden für den Prototyp zwei Classifizier benötigt. Für unseren Anwendungsfall soll das Modell vorhersagen können, wie wahrscheinlich es sich beim Input um ein SNOMED CT Code vom Typ «procedure» und «body structure» handelt. Daher werden die Classifizier vom BERT-Modell auf SNOMED CT Konzepte vom Typ «procedure», «body structure» und «disorder» mittels Supervised Learning trainiert. Konzepte vom Typ «disorder» deshalb, um einen ausgeglichenen Datensatz zu erhalten. Damit das überwachte Lernen mit ungefähr gleich vielen «korrekten» SNOMED CT Codes und nicht «korrekten» SNOMED CT Codes durchgeführt werden kann.

**Hinweis:** Wenn in den nachfolgenden Ausführungen und Abbildungen die Begriffe «SNOMED CT» und «Not SNOMED CT» im Zusammenhang mit dem Finetuning der Classifizier beschrieben werden, sind mit SNOMED CT immer Konzepte vom Typ «procedure» und «body structure» gemeint. Der Begriff Not SNOMED CT steht für alle Konzepte des Typen «disorder».

Für den ersten als auch für den zweiten Classifizier wird auf Basis vom SNOMED CT International Release Package eine Tabelle mit allen Konzepten erstellt.

Weiter werden die vom SIWF bereits parametrisierten Facharztztitel manuell durch uns in einer Tabelle zusammengetragen. Nicht jede Prozedur wurde bereits parametrisiert. Daher wird der Datensatz bereinigt. Es werden nur die Prozeduren aufgenommen, bei welchen mind. ein Konzept hinterlegt ist. Diese werden zudem noch mit DeepL ins Englische übersetzt. Prozeduren, bei welchen mehrere Konzepte hinterlegt sind, werden dupliziert und mit je einem vorhandenen SNOMED CT Konzept in der Tabelle eingetragen.

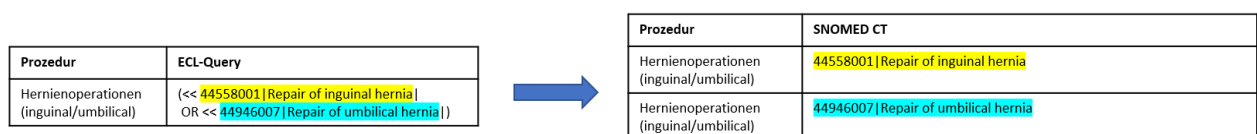


Abbildung 16: Umgang >1 Konzepte pro Prozedur

Die Tabelle aus dem SNOMED CT Release Package und dem SIWF werden zu einem Datensatz zusammengefügt. Die Bezeichnungen der SNOMED CT Konzepte aus dem Releasepackage beinhalten teilweise die Bezeichnung des Typs in Klammern. Beispiel: Excision of lesion of patella (procedure). Um jedoch die Modelle robuster zu machen werden in einem nachgelagerten Schritt die Typbezeichnungen

aus den Einträgen gelöscht. Weiter wird im Preprocessing der erstellte Datensatz für das Supervised Learning in zwei Tabellen geteilt.

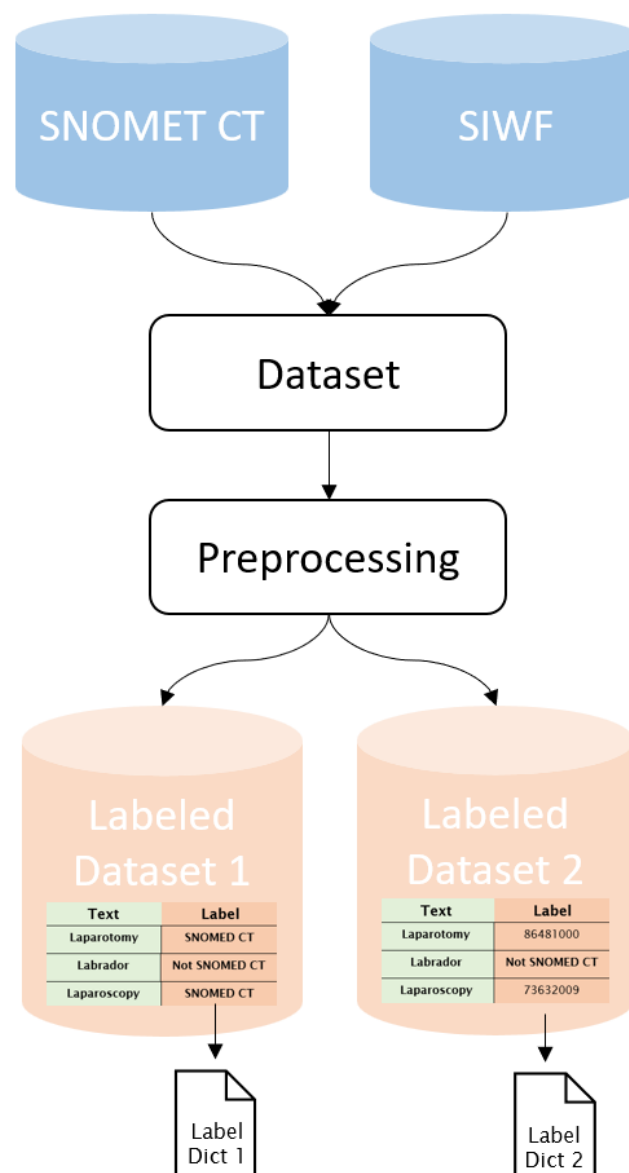


Abbildung 17: Workflow Data collection

Die erste Tabelle (Labeled Dataset 1) besteht aus zwei Spalten (Text und Label) und beinhaltet den Text und das dazugehörige Label SNOMED CT oder Not SNOMED CT. Die zweite Tabelle, der «Labeled Dataset 2» beinhaltet den Text und als Label den dazugehörigen SNOMED CT Code. Aus beiden gelabelten Datensätzen wird ein separates Verzeichnis (engl. Dictionary) der enthaltenen Labels erstellt. Dieses wird für das Supervised Learning benötigt.

Das Verzeichnis für den ersten Classifier beinhaltet zwei Labels (SNOMED CT und Not SNOMED CT) und wird als Label Dict 1 bezeichnet. Das Verzeichnis für den zweiten Classifier beinhaltet hingegen über 89.000 Labels. Denn jeder SNOMED CT Code vom Typ «procedure» und «body structure» wird als eigenes Label im Verzeichnis aufgenommen. Dies wird als unser Label Dict 2 bezeichnet. Zudem wird beim «Labeled Dataset 2» auf alle Prozeduren des Typs «disorder» verzichtet, weil beim Modell 2 auf die einzelnen Labels gemappt wird und nicht wie bisher auf die zwei Kategorien (SNOMED CT oder Not SNOMED CT). Dies führt dazu, dass die einzelnen Labels für das Training unter-repräsentiert sind und daher das Dataset 2 vergrößert werden muss. Nachfolgende Abbildung fasst das aufgeführte Vorgehen zusammen.

## Supervised Learning

Im zweiten Teil des Finetunings erfolgt die Durchführung des Trainings mittels überwachtem Lernens (Supervised Learning). Das Finetuning der Classifier wird einerseits über eine NVIDIA DGX Workstation der BFH Server realisiert und andererseits mit Google Colab. Google Colab erlaubt die Verknüpfung mit Google Drive und verfügt über eine Versionskontrolle, zudem können bestehende Modelle via Hugging Face geladen und verwendet werden.

Für den ersten Classifier (Modell 1) wird der Trainingsdatensatz 1 (Dataset 1) und das dazugehörige Verzeichnis (Label Dict 1) verwendet. Für den zweiten Trainingsdatensatz entsprechend das Dataset 2 und Label Dict 2. Diese werden anschliessend in die Umgebung geladen (Load Dataset and Dictionary). Für das Training wird das Dataset in Trainings-, Test- und Validierungsdatensatz randomisiert aufgeteilt.

Würde das Modell mit dem ganzen Datensatz trainiert werden, lernte es einfach die Texte, die es gerade gesehen hat und hätte bei der Evaluation ein perfektes Ergebnis. Trotzdem könnte es bei vorher nie gesehenen Texten nichts Brauchbares als Output generieren. Diese Situation wird auch als Overfitting bezeichnet. Um dies zu vermeiden ist es gängige Praxis, bei der Durchführung von Supervised Learning den Trainingsdatensatz wie beschrieben aufzuteilen. Als Beispiel wird hier der gesplittete (test, train & val) Datensatz vom Modell 1 aufgeführt:

Label	label_code	data_type	
NOT SNOMED CT	1	test	22267
		train	178745
		val	22118
SNOMED CT	0	test	24184
		train	192855
		val	24332

Abbildung 18: Dataset 1 für Training

Weiter wird der Tokenizer vom vortrainierten BERT-Modell geladen. Dies um die einzelnen Texteinträge des Train-, Val und Testsets zu tokenisieren. Beim Tokenisieren wird der Ursprungstext in sogenannte Tokens aufgeteilt und anschliessend in numerische Werte konvertiert (Word Embeddings), welche die Wörter darstellen.

Der Tokenizer codiert den Input, mit einer maximalen Länge von 512 Tokens, in die notwendige Form für BERT (engl. encode) und retourniert einen Tensor (Multidimensionales Array). Für beide Modelle wird eine Instanz der Klasse «BertForSequenceClassification» aus dem vortrainierten BERT-Modell erstellt. Ein Modell mit zwei Labels und das andere Modell mit über 89'000 Labels.

Weiter wird pro Datensatz eine DataLoader Instanz benötigt. Der DataLoader verpackt den Datensatz in Batches. Um den Datensatz zufällig auf die Batches zu verteilen, nutzt der DataLoader einen RandomSampler. Diese zufällige Aufteilung wird jedoch nur im Trainingsdatensatz gemacht, da nur in dieser Phase die Gewichte angepasst werden. Dadurch wird gewährleistet, dass das Modell nur aus zufälligen Daten lernt und keine Rückschlüsse aus der Reihenfolge ziehen kann.

Der Dataloader liefert nun eine iterierbare Ansammlung von Batches, welche wiederum auch wieder iteriert werden können. Ebenfalls wird für das Supervised Learning ein Optimizer benötigt. Ein Optimizer ist ein Algorithmus, der die Attribute des neuronalen Netzes, wie Gewichte und Lernrate, verändert. So hilft er u.a. die Genauigkeit zu verbessern.

All die aufgeführten Schritte werden in den Prozess «Training preparation» zusammengefasst. Weiter werden die Leistungsmerkmale (F1-Score und Accuracy) berechnet. Damit kann nach jeder durchlaufenen Trainingsepoche, das Modell evaluiert (Evaluation) werden. Schlussendlich wird das erfolgreichste Modell gespeichert (Save Best Model). Nachfolgende Abbildung fasst das Supervised Learning für das Modell 1 zusammen.

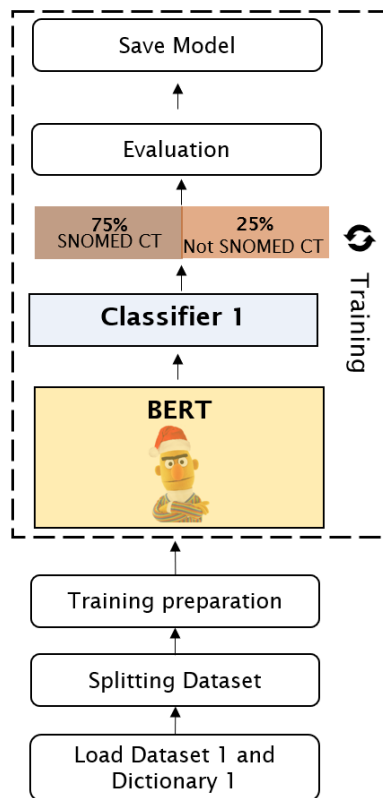


Abbildung 19: Workflow Supervised Learning Modell 1

Nachfolgend wird der oben aufgeführte Workflow (Abbildung 19) für die Data collection und das Supervised Learning für beide Modelle zusammenfassend dargestellt. Die Abbildung 20 wurde inspiriert vom Blog von Jay Alamar (7):

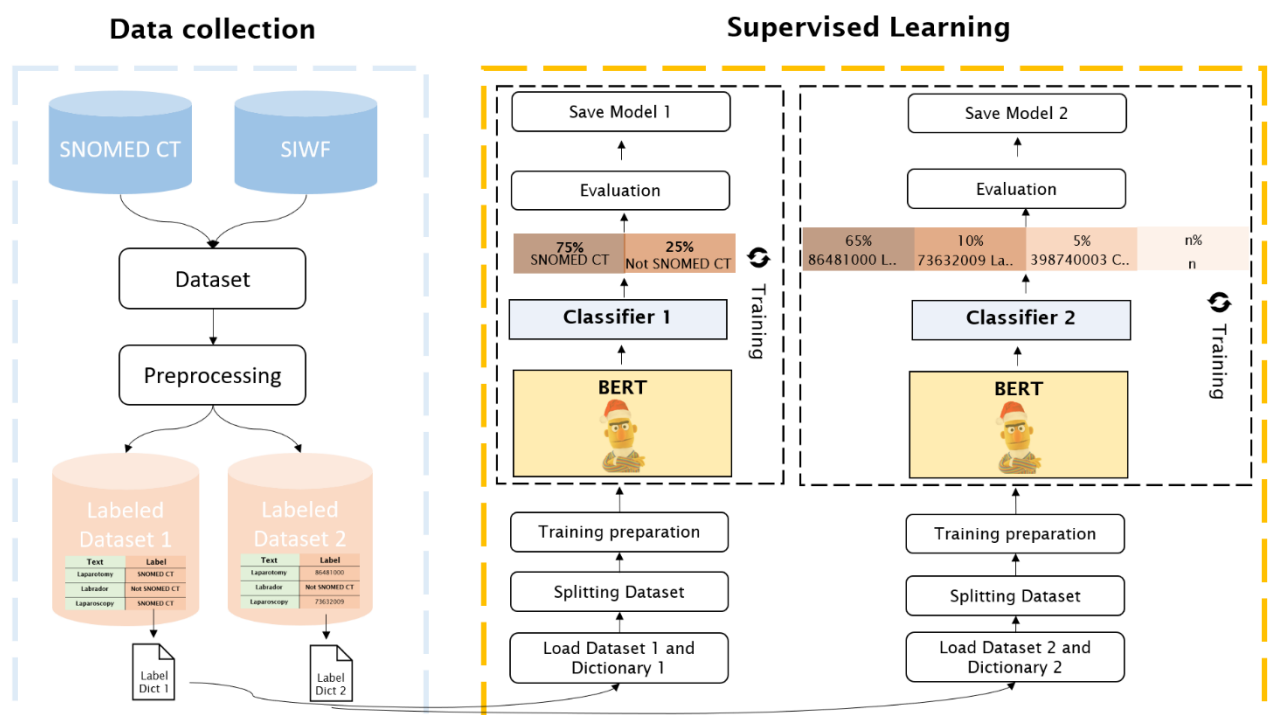


Abbildung 20: Workflow Bereich Finetuning

### 3.4.2.3 Bereich Umsetzung

Für die Erarbeitung und Darstellung der angestrebten Systemarchitektur werden zwei unterschiedliche Systemdiagramme erstellt. Die zwei Bereiche, die abgedeckt werden, sind:

- Verwendung Desktopapplikation lokal
- Verwendung Desktopapplikation mit Server

Die Weiterentwicklung der Modelle, respektive die Umsetzung des Finetunings, wird im Bereich Finetuning genauer erläutert und ist nicht direkter Bestandteil der Umsetzung.

Ziel ist eine klare Übersicht über die zu verwendenden Tools (Programmiersprache, Libraries, Daten, Speicherorte) und eine grobe Darstellung des Datenflusses zwischen den Systemen darzustellen.

Zudem wird anhand des Pflichtenheftes und der Systemdiagramme das initiale Mock-up erstellt.

### 3.4.2.4 Bereich Gesamtevaluation

Innerhalb des Bereichs der Gesamtevaluation im Detailkonzept sollen die Evaluationskriterien einerseits für die Modelle und andererseits für die Desktopapplikation erarbeitet und beschrieben werden. Die im Detailkonzept erarbeiteten Vorgehensweise soll uns ermöglichen die obengenannten Kategorien einzeln zu bewerten und zu evaluieren.

#### Modelle

Die Evaluation der weiterentwickelten BERT-Modelle lässt sich in zwei Bereiche aufteilen. Der erste Bereich bezieht sich auf die Leistungsmerkmale eines Modelles. Diese geben Aufschluss über die Qualität des entwickelten Classifiers und werden direkt nach dem Training eines Modelles gemessen (siehe Bereich Finetuning).

Der andere Bereich ist eine Visualisierung für die Einschätzung des verwendeten BERT Modelles bzw. dessen Word Embeddings. Für die Evaluation der Leistungsmerkmale eines Modells wird eine Confusion Matrix verwendet. Daraus lassen sich die Merkmale Recall, Precision, Accuracy und F1 ableiten (27) .

		<i>gold standard labels</i>		
		gold positive	gold negative	
<i>system output labels</i>	system positive	true positive	false positive	precision = $\frac{tp}{tp+fp}$
	system negative	false negative	true negative	
		recall = $\frac{tp}{tp+fn}$		accuracy = $\frac{tp+tn}{tp+fp+tn+fn}$

Abbildung 21: Darstellung einer Confusion Matrix (27)

Für die Umsetzung im Code wird die Python Library «scikit-learn» verwendet. In unserer Umsetzung handelt es sich um einen Multiclass Classifier. Der Output der Classifier gibt mittels Softmax-Funktion eine Wahrscheinlichkeit der Zugehörigkeit des Input zu einer Klasse aus. Dies führt dazu, dass für jede Klassen eine Wahrscheinlichkeit ausgegeben wird.



Um den Output messen zu können, haben wir uns entschieden, dass jeweils die Klasse mit der höchsten Wahrscheinlichkeit für die Evaluation verwendet wird. Dadurch ergeben sich nachfolgende Tabelle 7 mit den möglichen Resultaten (Metriken) für die Berechnung der Leistungsmerkmale.

Metrik	Beschreibung	Beispiel
False Negative (FN)	Vom Modell vorhergesagte Klassifikation mit der höchsten Wahrscheinlichkeit entspricht fälschlicherweise (false) einem NOT SNOMED CT Code (negative).	<p>Textinput: «Reductions (fractures, luxations) » (procedure)</p> <p>Korrekter Output wäre:  Model 1: SNOMED CT  Model 2: 86052008 Closed reduction of fracture</p> <p><b>False Negative:</b>  Model 1: NOT SNOMED CT  Model 2: 89555 none of the trained concepts</p>
False Positive (FP)	Vom Modell vorhergesagte Klassifikation mit der höchsten Wahrscheinlichkeit entspricht fälschlicherweise (false) einem SNOMED CT Code (positive).	<p>Textinput: «Stenosis of Jejunum» (disorder)</p> <p>Korrekter Output wäre:  Model 1: NOT SNOMED CT  Model 2: 89555 none of the trained concepts</p> <p><b>False Positive:</b>  Model 1: SNOMED CT  Model 2: 86052008 Closed reduction of fracture</p>
True Negative (TN)	Vom Modell vorhergesagte Klassifikation mit der höchsten Wahrscheinlichkeit entspricht dem geprüften Text ( <b>NOT</b> SNOMED CT).	<p>Textinput: «Stenosis of jejunum» (disorder)</p> <p><b>True Negative = korrekter Output:</b>  Model 1: NOT SNOMED CT  Model 2: 89555 none of the trained concepts</p>
True Positive (TP)	Vom Modell vorhergesagte Klassifikation mit der höchsten Wahrscheinlichkeit entspricht dem geprüften Text (SNOMED CT).	<p>Textinput: «Reductions (fractures, luxations) » (procedure)</p> <p><b>True Positive = korrekter Output:</b>  Model 1: SNOMED CT  Model 2: 86052008 Closed reduction of fracture</p>

Tabelle 7: Beschreibung Metriken der Leistungsmerkmale

In nachfolgender Tabelle 8 werden die erwähnten Leistungsmerkmale für die Evaluation der weiterentwickelten Modelle kurz zusammengefasst:

Leistungsmerkmal	Beschreibung	Formel
Accuracy	Gibt das Verhältnis der richtigerweise als richtig erkannt und richtigerweise als falsch erkannten Labels im Rahmen aller Daten des Datensatzes an. Dies ist ein Mass für die Genauigkeit bezüglich der Vorhersage der richtigen Klassifizierung.	$\frac{TP + TN}{TP + FP + TN + FN}$
F1	Beschreibt die Gesamtgenauigkeit der Vorhersagen. Dazu werden Precision und Recall gleich gewichtet.	$\frac{2 \times Precision \times Recall}{Precision + Recall}$
Precision	Gibt das Verhältnis der richtigerweise als richtig erkannten zu den richtig erkannten und falsch erkannten Labels an. Somit gibt die Precision die Genauigkeit der Vorhersage hinsichtlich der falsch-positiven Klassifizierungen an.	$\frac{TP}{TP + FP}$
Recall	Gibt das Verhältnis der richtigerweise als richtig erkannten Labels zu den richtig erkannten und fälschlicherweise nicht erkannten Labels an. Somit gibt der Recall die Genauigkeit der Vorhersage hinsichtlich der falsch-negativen Klassifizierungen an.	$\frac{TP}{TP + FN}$

Tabelle 8: Beschreibung Leistungsmerkmale bei der Modellevaluation

Für die Evaluation der Modelle wird der Trainingsdatensatz in einen Trainings- und Testdatensatz aufgeteilt. Anhand des Testdatensatzes werden nach dem Training die Leistungsmerkmale Recall, Precision, Accuracy und F1 automatisch berechnet. Zusätzlich werden die jeweiligen Datensätze vor der Verwendung nochmals aufgeteilt, um einen Evaluationsdatensatz zu erhalten. Dieser wird während des Trainings für die Validierung des Trainings bzw. des Testes benötigt. Dies dient der Vermeidung und der Erkennung von Overfitting des Modelles.

## Prototyp

Die Evaluation des Prototyps umfasst die Desktopapplikation und die Verwendung der Modelle. Zum einen soll die Erwartungshaltung vom SIWF zu der Benutzeroberfläche der Desktopapplikation abgeholt werden und zum anderen eine Vorgehensweise erarbeitet werden, die sicherstellt, dass das Modell einen wertvollen Beitrag für das SIWF leistet. Um dies zu überprüfen, wird der Prototyp nach der Fertigstellung mit dem SUS Score durch die zwei Personen beim SIWF evaluiert. Nachfolgend werden die Fragen vom SUS Score in deutscher Sprache tabellarisch (Tabelle 9) mit der entsprechenden Punktzahl aufgeführt (28). Dank dieser Fragen kann schnell und mit wenig Aufwand die Benutzerfreundlichkeit einer Applikation erfasst werden.

Nr.	Aussage	Mögliche Antworten				
		Stimme nicht zu	Stimme eher nicht zu	Neutral	Stimme eher zu	Stimme zu
1	Ich kann mir sehr gut vorstellen, das System regelmässig zu nutzen.	0	1	2	3	4
2	Ich empfinde das System als unnötig komplex.	4	3	2	1	0
3	Ich empfinde das System als einfach zu nutzen.	0	1	2	3	4
4	Ich denke, dass ich technischen Support brauchen würde, um das System zu nutzen.	4	3	2	1	0
5	Ich finde, dass die verschiedenen Funktionen des Systems gut integriert sind.	0	1	2	3	4
6	Ich finde, dass es im System zu viele Inkonsistenzen gibt.	4	3	2	1	0
7	Ich kann mir vorstellen, dass die meisten Leute das System schnell zu beherrschen lernen.	0	1	2	3	4
8	Ich empfinde die Bedienung als sehr umständlich.	4	3	2	1	0
9	Ich habe mich bei der Nutzung des Systems sehr sicher gefühlt.	0	1	2	3	4
10	Ich musste die Hilfe der Entwickler beanspruchen, bevor ich mit dem System arbeiten konnte.	4	3	2	1	0

Tabelle 9: Übersicht Fragen SUS Score

Am Schluss wird die Punktzahl aufaddiert, durch die Anzahl teilnehmenden Personen dividiert und mit 2.5 multipliziert. Dies ergibt den SUS Score. Die Interpretation des Resultats erfolgt nach untenstehender Skala (Abbildung 22).

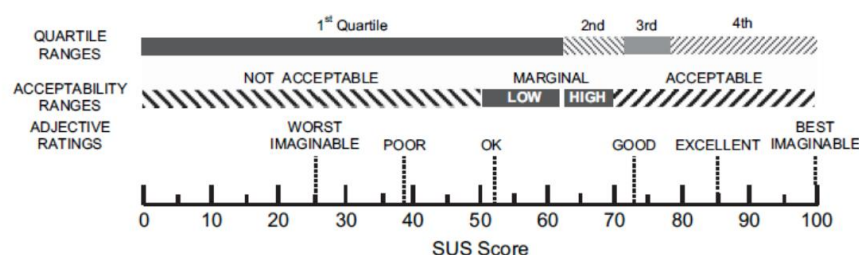


Abbildung 22: System Usability Scale (28)

## 4 Ergebnisse

In diesem Kapitel werden die Ergebnisse dieser Arbeit vorgestellt. Im ersten Abschnitt werden die Resultate aus den Variantenanalysen für den gefällten Technologieentscheid präsentiert. Daraufhin werden die während des Trainings aufgezeichneten Ergebnisse der entwickelten Modelle aufgeführt. Weiter wird die grafische Benutzeroberfläche vom Prototyp dargelegt und im letzten Abschnitt dieses Kapitels werden die Ergebnisse aus der Evaluation (F1 Wert und SUS-Score) vorgestellt.

### 4.1 Technologieentscheid

Der Technologieentscheid beinhaltet drei Variantenanalysen. Zwei Variantenanalysen wurden innerhalb der Domäne «NLP Pipeline» durchgeführt und fließen direkt in den Bereich vom Finetuning.

#### Bereich Finetuning (NLP Pipeline)

Modell	BERT	BioBERT <sup>1</sup>	ClinicalBERT <sup>1</sup>	ClinicalBioBERT <sup>2</sup>	EhrBERT <sup>2</sup>
Trainiert auf	800 Mio. Wörter aus dem BooksCorpus und 2'500 Mio. Wörter vom englischen Wikipedia (14).	4.5 Mia. Wörter PubMed Abstracts und 13.5 Mia. Wörter Artikel PMC (29).	2 Mio. medizinischen Notizen aus dem MIMIC-III Datensatz (30).	2 Mio. medizinischen Notizen aus dem MIMIC-III Datensatz (30) und NOTEEVENTS Tabelle aus MIMIC-III mit 880 Mio. Wörter (31).	1.5 Mio. elektronischen Krankenakten (32).
<sup>1</sup> basiert auf BERT, <sup>2</sup> basiert auf BioBERT					
Verfügbarkeit	3	3	3	3	0
3 = Ja 0 = Nein					
State-of-the-art	1	3	2	0	1
1 = gering 2 = mittel 3 = hoch					
<b>Summe</b>	<b>4</b>	<b>6</b>	<b>5</b>	<b>3</b>	<b>1</b>

Tabelle 10: Ergebnis Variantenanalyse BERT-Modell

Tools zur Visualisierung der Gütekriterien	Plotly	TensorBoard Projector
Dokumentation	2	2
Flexibilität	1	3
1 = gering 2 = mittel 3 = hoch		
Einarbeitungsaufwand	1	2
1 = hoch 2 = mittel 3 = gering		
Interaktivität	0	3
Kompatibilität Colab	3	3
3 = Ja 0 = Nein		
<b>Summe</b>	<b>7</b>	<b>13</b>

Tabelle 11: Ergebnis Variantenanalyse Tools zur Visualisierung der Gütekriterien

### Bereich Umsetzung

Da für den Bereich Umsetzung zwei Varianten berücksichtigt wurden. Die lokale Verwendung des BERT-Modells und die dezentral via Server. Für das passende Micro Framework der zweiten Variante ist eine Analyse durchgeführt worden. Das Ergebnis wird nachfolgend präsentiert.

<i>Micro Framework</i>	FastAPI	Tornado	Flask	Sanic	Falcon	Bottle	Hug	Eve
Dokumentation	1	1	3	2	2	2	2	2
Popularität	3	2	3	2	2	2	1	2
1 = gering 2 = mittel 3 = hoch								
Einarbeitungsaufwand	3	2	3	2	1	2	1	2
Installationsaufwand	3	2	3	2	1	2	1	2
1 = hoch 2 = mittel 3 = gering								
<b>Summe</b>	<b>10</b>	<b>7</b>	<b>12</b>	<b>8</b>	<b>6</b>	<b>8</b>	<b>5</b>	<b>8</b>

Tabelle 12: Ergebnis Variantenanalyse Server Framework

Aufgrund der oben aufgeführten Resultate ist folgender Technologieentscheid gefällt worden:

- BERT-Modell: **BioBERT**
- Tool zur Visualisierung der Gütekriterien: **TensorBoard Projector**
- Server Framework: **Flask**

## 4.2 Bereich Finetuning

Im Kapitel Methodik wurden die zwei grundsätzlichen Teilaufgaben für den Bereich Finetuning bereits erwähnt. Um nachfolgende Ergebnisse besser einordnen zu können, werden diese hier nochmals aufgeführt.

- Datenaufbereitung für das Supervised Learning (Data collection)
- Durchführung des Trainings (Supervised Learning)

Details über die technische Umsetzung des Finetunings sind in der technischen Dokumentation im Anhang 10.7 aufgeführt. Innerhalb der Realisierungsphase haben wir uns dazu entschieden einen dritten Multiclass Classifier (Modell 3) auf Basis von BioBERT zu entwickeln. Die Motivation hinter diesem Entscheid wird im Kapitel Diskussion ausgeführt. Das Ergebnis der Datenaufbereitung (Data collection) für das Training wird in nachfolgender Tabelle dargestellt. Die Resultate von der Durchführung des Trainings werden in den Kapitel 4.2.2 und 4.2.3 aufgeführt.

Merkmale	labeled_dataset1 (Modell 1)	labeled_dataset2 (Modell 2)	labeled_dataset3 (Modell 3)
Anzahl Texte	464'501	241'403	2'473
Anzahl Tokens pro Text (Mittelwert)	5.3	7.1	43.7
Anzahl Tokens gesamt	2'472'080	1'708'809	108'110

Tabelle 13: Übersicht erstellter Trainingsdatensätze

Wie in der Abbildung 17 «Workflow Data collection» dargestellt, wird aus den gelabelten Datensätze einen Labelverzeichnis erstellt.

Merkmale	label_dict_m1 (Modell 1)	label_dict_m2 (Modell 2)	label_dict_m2 (Modell 3)
Anzahl Labels	2	89855	368

Tabelle 14: Übersicht erstellter Labelverzeichnisse

#### 4.2.1 Modell

In diesem Kapitel werden die Merkmale der entwickelten Modelle als Ergebnisse präsentiert. Die «model – files» und die dazugehörigen Labelverzeichnisse stellen wir auf Anfrage zur Verfügung. Die Tabelle 15 fasst die wichtigsten Informationen als Ergebnisse von den Modellen zusammen.

Merkmale	Modell 1	Modell 2	Modell 3
Beschreibung	Ein BioBERT-Modell mit einem binären Klassifikator. Es berechnet anhand des Inputs, ob es sich um ein Konzept der Kategorie 'procedure' und/oder 'body structure' (SNOMED CT) oder der Kategorie 'disorder' (NOT SNOMED CT) handelt.	Ein BioBERT-Modell mit einem Multiclass Klassifikator. Das Modell 2 umfasst 89'855 Klassen. Bis auf eine Klasse, repräsentiert jede Klasse ein SNOMED CT Konzept der Kategorie 'procedure' oder 'body structure'. Das Modell 2 berechnet anhand des Inputs, die n wahrscheinlichsten Konzepte aus den 89'855 Klassen.	Ein BioBERT-Modell mit einem Multiclass Klassifikator. Das Modell 3 umfasst 368 Klassen. Bis auf eine Klasse, repräsentiert jede Klasse ein SNOMED CT Konzept der Kategorie 'procedure' oder 'body structure', die bereits vom SIWF in den Weiterbildungsprogrammen identifiziert wurden. Das Modell 3 berechnet anhand des Inputs, die n-Anzahl wahrscheinlichsten Konzepte aus den 368 Klassen.
Anzahl Klassen	2	89'855	368
Finetuning-Umgebung	BFH NVIDIA DGX Station™ A100, 1 von 4 Tesla V100-DGXS-32GB GPU	Google Colab Pro+ NVIDIA P100 16GB oder Tesla T4 16GB	Google Colab Pro+ NVIDIA P100 16GB oder Tesla T4 16GB
Trainingsdauer	24 Std.	264 Std. / 11 Tage	3.3 Std.
Grösse Trainingsdaten [Anzahl Text Einträge]	464'501	241'403	2'473
Batchgrösse	8	16	16

Tabelle 15: Übersicht Merkmale Modelle

#### 4.2.2 Validation F1 und Test F1

Innerhalb des Finetunings der drei Modellen ist anhand des Validierungsdatensatzes während des Trainings der F1 Wert (engl. Validation F1) berechnet worden. Der Validierungsdatensatz ist eine Stichprobe von 12.5 % aus dem Trainingsdatensatz, welche vor dem Start genommen wird.

Dank dem Validation F1 kann während des Finetunings und nach jeder Epoche eine Aussage gemacht werden, wie gut das Modell Vorhersagen auf der Basis von unbekannten Daten macht. Durch die Verwendung der immer gleichen Daten bei jeder Epoche ist eine Entwicklung ersichtlich.

Nebst dem Validierungsdatensatz wird ein Testdatensatz erstellt. Hierfür wird ebenfalls eine vor dem Start entnommene Stichprobe von 12.5 % vom Trainingsdatensatz angewendet. Die Auswertung vom Testdatensatz erfolgt nach Abschluss des Finetunings und gibt Aufschluss darüber, wie «robust» das Modell nach Abschluss des Trainings ist.

Es folgen die Verläufe des Validation F1 Wertes der drei Modelle über 100 Epochen, wobei das Training vom Modell 1 bereits nach fünf Epochen und das vom Modell 2 nach 97 Epochen beendet wurde.

Merkmal	Modell 1	Modell 2	Modell 3
	%	%	%
Validation F1	99.81	20.97	72.76
Test F1	N.A.	20.74	62.95

Tabelle 16: Validation und Test F1

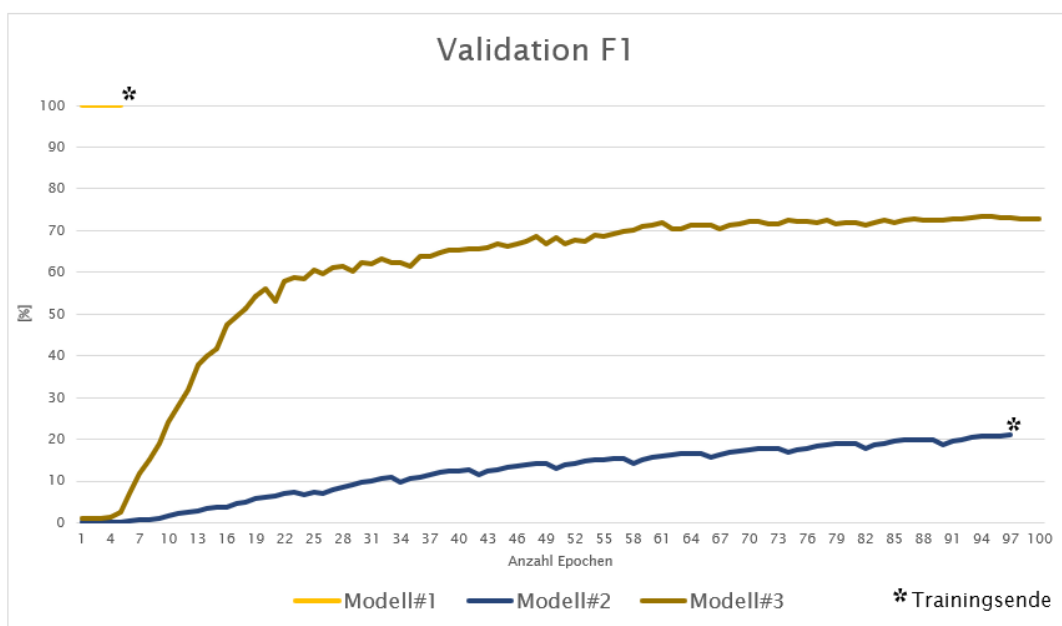


Abbildung 23: Validation F1 – Liniendiagramm



#### 4.2.3 Training loss und Validation loss

Während des Trainings wird zudem der Trainingsverlust (engl. Train loss) und Validierungsverlust (engl. Val loss) erhoben. Training loss gibt an, wie gut sich das Modell innerhalb des Trainings dem Trainingsdatensatz anpasst. Hingegen gibt der Validation loss an, wie gut das Modell an neue Daten angepasst wird. Nachfolgende Abbildungen 24, 25 und 26 zeigen den Verlauf vom Training und Validation loss der Modelle 1, 2 und 3.

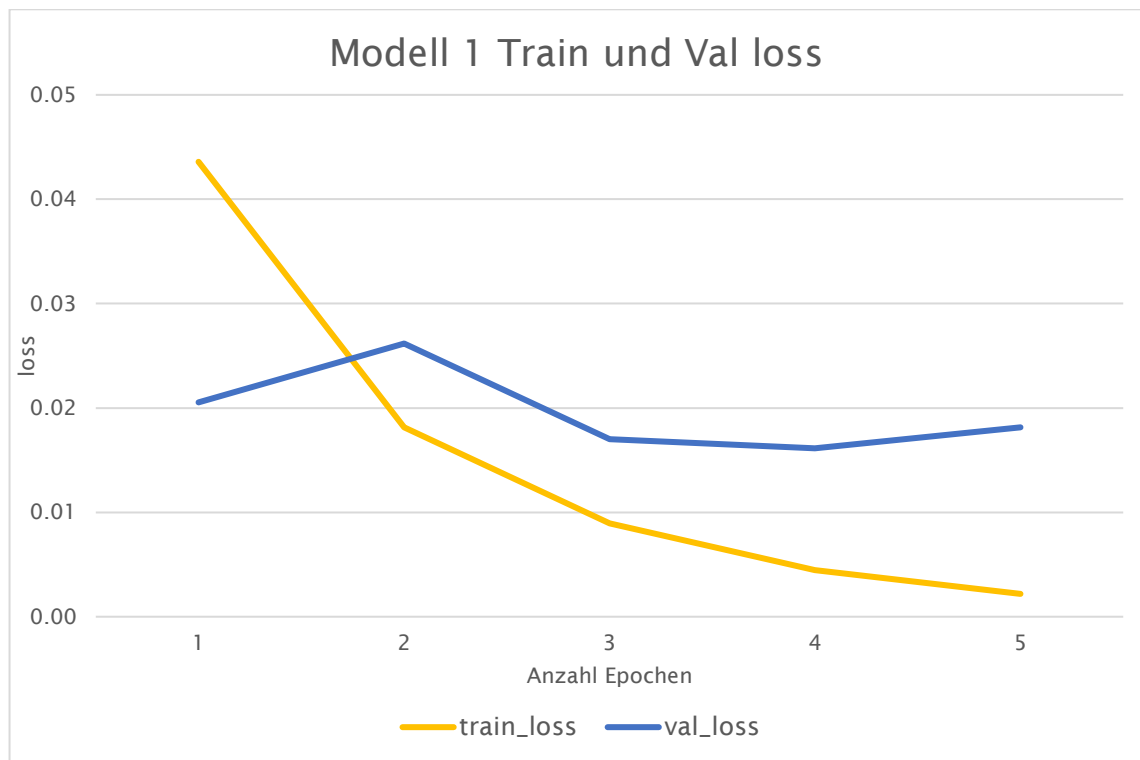


Abbildung 24: Modell 1 Train und Val loss

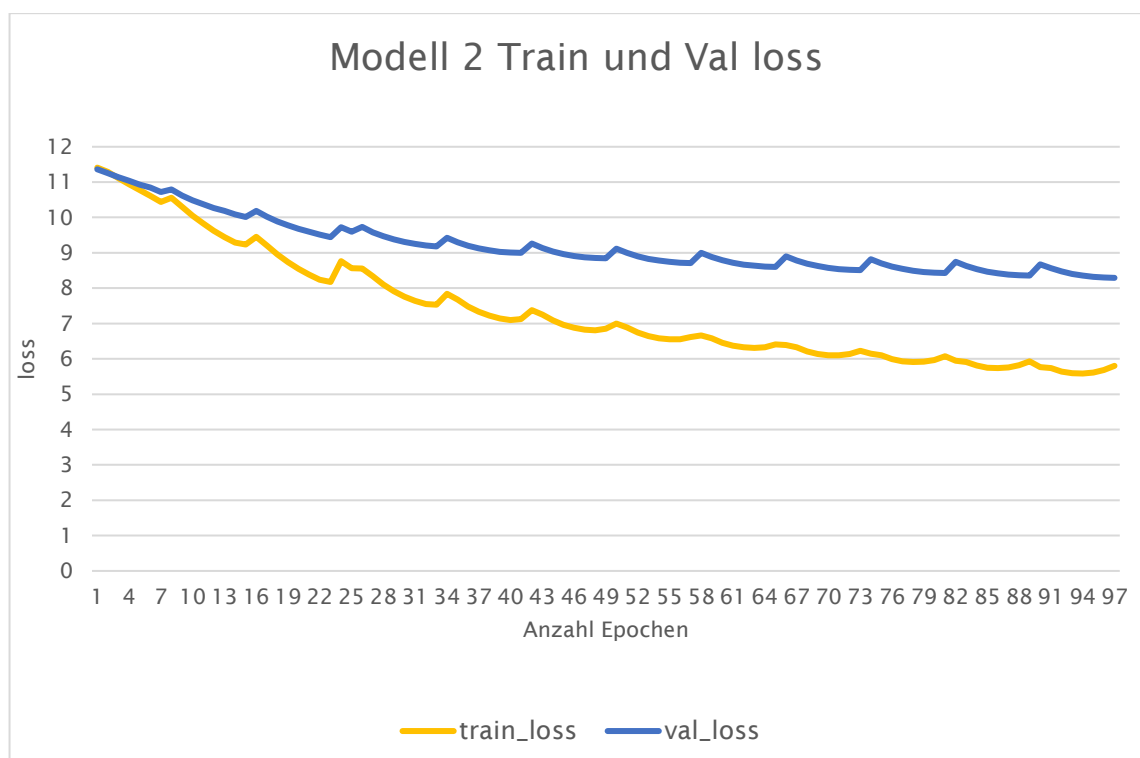


Abbildung 25: Modell 2 Train und Val loss

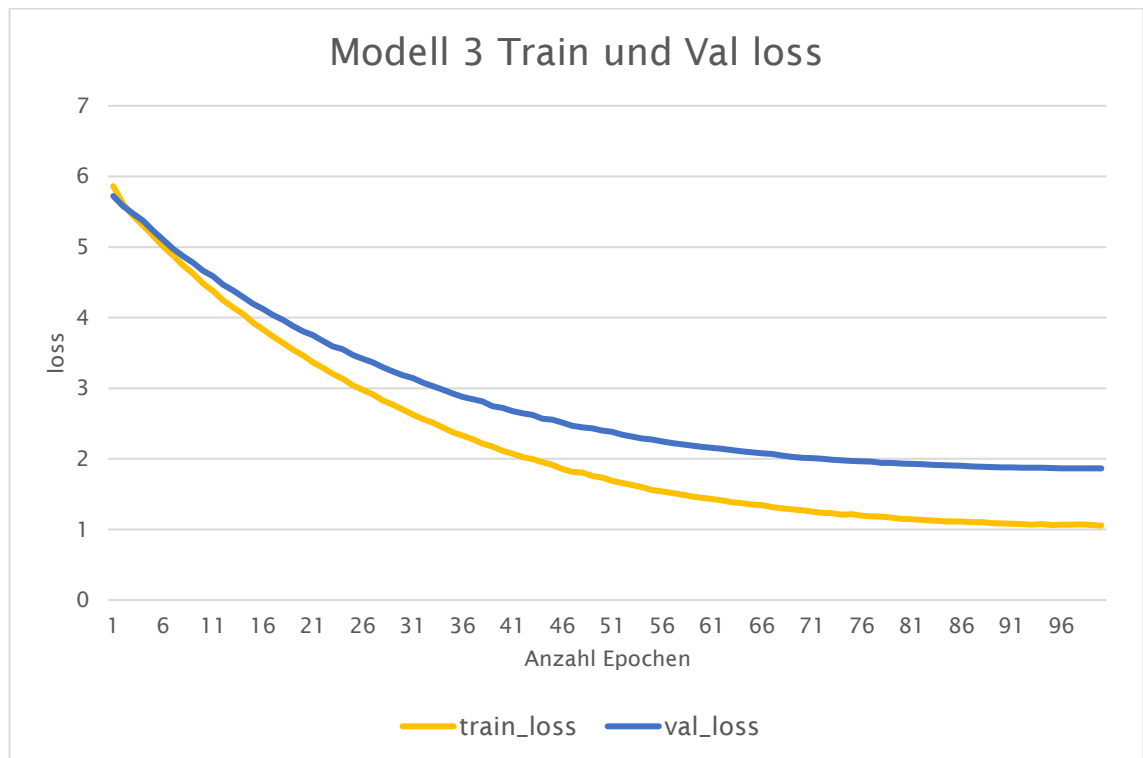


Abbildung 26: Modell 3 Train und Val loss

### 4.3 Bereich Umsetzung

Für die Desktopapplikation wurden zwei Systemdiagramme entworfen. Einerseits für die lokale Verwendung der im Bereich Finetuning weiterentwickelten BERT-Modellen und andererseits für die Verwendung der BERT-Modelle mittels Server.

Die Desktopapplikation hat mehrere Bestandteile: Ein GUI, welches mit Hilfe von PyQt5 umgesetzt wird, die BERT-Classfier, welche durch Finetuning erstellt wurden und die Businesslogik für die Verwendung der Modelle. Für die Bewirtschaftung und Anpassung der Software auf die lokalen Begebenheiten der Computer wird innerhalb der Software eine Konfigurationsdatei (config file) erstellt. Dadurch lässt sich die Applikation auch mit neuen Modellen erweitern. Für die Verwendung der Applikation ist der Zugang zum Internet notwendig. Nachfolgend wird die erste Variante des Systemdiagramms «Desktopapplikation lokal» präsentiert.

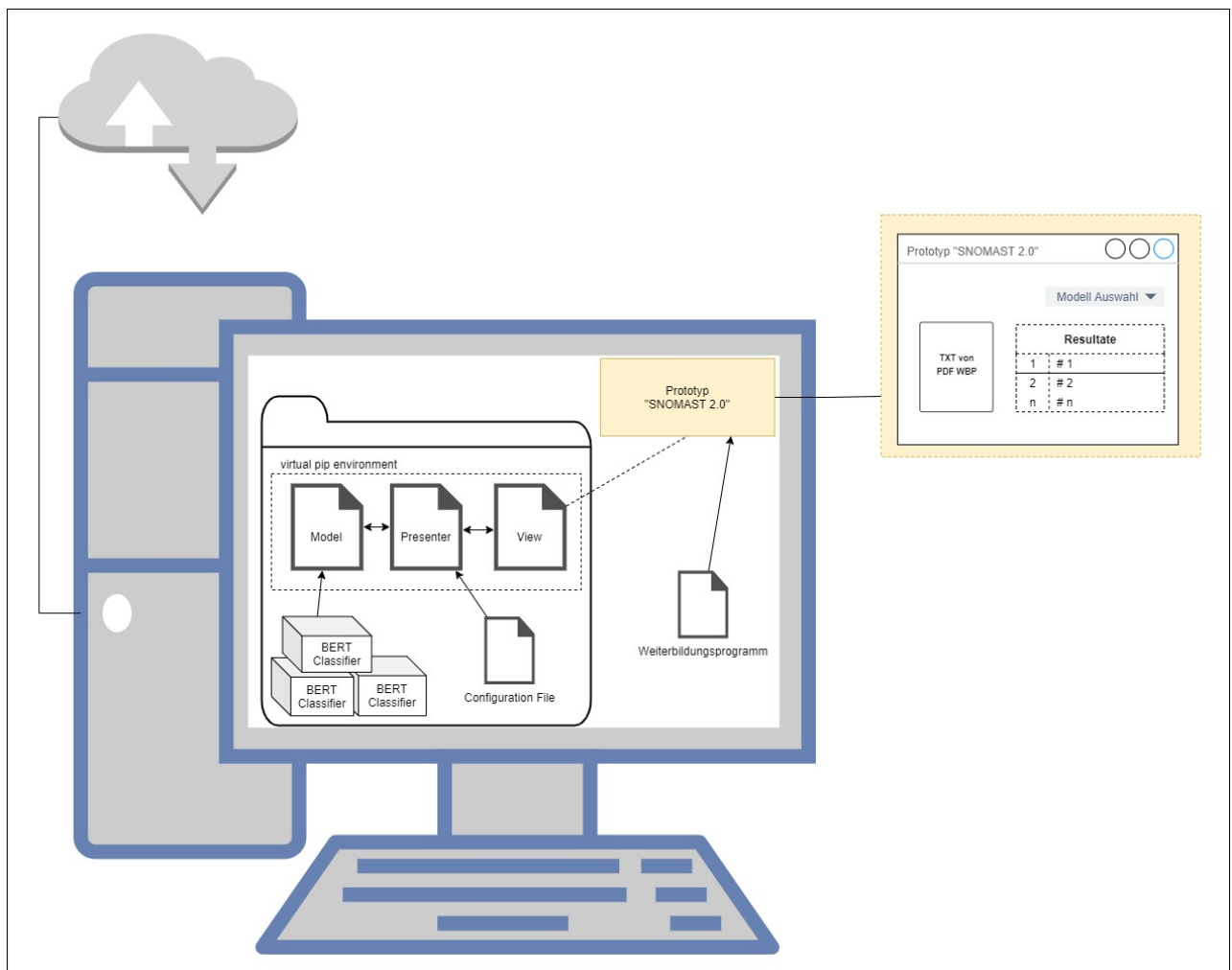


Abbildung 27: Systemdiagramm «Desktopapplikation lokal»

Für die Umsetzung der Applikation bietet sich das Model-View-Presenter (MVP) Entwurfsmuster an. Damit könnte der Prototyp für die funktionale KANN-Anforderung (ID: 8.FA – BERT-Modell kann via einer REST-API genutzt werden). so angepasst werden, dass auf dem lokalen Rechner nur das GUI installiert wird. Das GUI würde über eine REST API mittels http-Nachrichten mit dem leicht angepassten Presenter/Businesslogik, die sich auf dem Server befindet, kommunizieren. Die Verwendung und Verwaltung der BERT-Classifer könnten somit auch auf den Server verlagert werden.

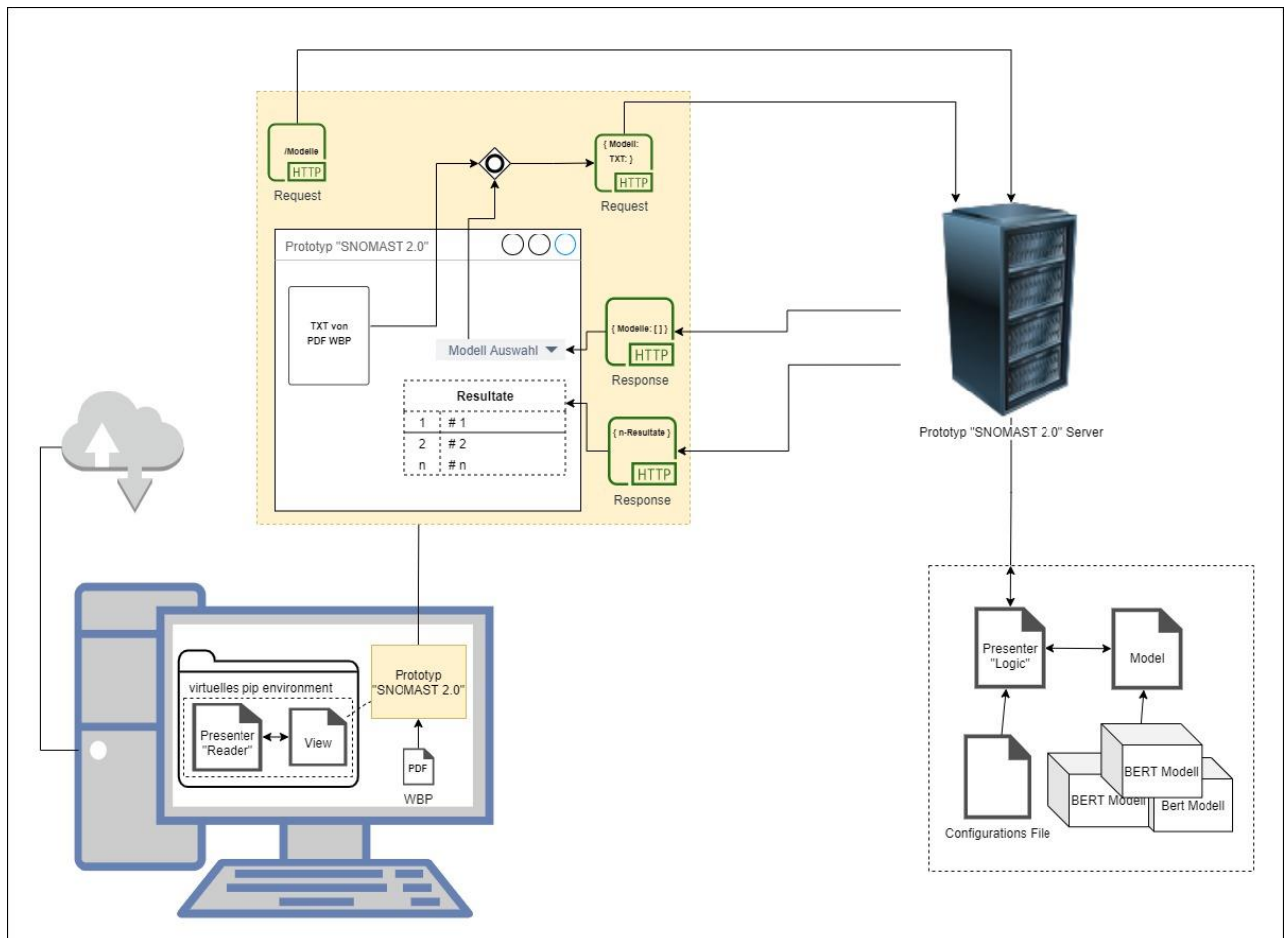


Abbildung 28: Systemdiagramm «Desktopapplikation mit Server»

#### 4.3.1 Prototyp

Die Umsetzung des Prototyps wurde im Rahmen des MVP (Model-View-Presenter) Patterns gestaltet. In diesem Softwareentwurfsmuster soll die View lediglich Daten von den Nutzerinnen und Nutzer sammeln und ihnen Daten anzeigen. Das Modell ist für die Verarbeitung der Daten zuständig und der Presenter ist die Schnittstelle zwischen der Ansicht (View) und des verarbeitenden Modells (Model). Die «View», «Logic», «Reader» und «Model» stellen im Prototyp eigenständige Python Dateien dar. Der Presenter ist in zwei Dateien aufgeteilt. Der «Reader» sind lediglich Hilfsmethoden für die «View», um die URL der zu importierenden Dateien auszulesen und die Texte in der «View» darzustellen. Hingegen ist «Logic» die Schnittstelle zwischen «View» und «Model». Details zum Prototyp sind in der technischen Dokumentation im Anhang 10.7 aufgeführt.

### 4.3.2 GUI

In diesem Kapitel wird das GUI von SNOMAST 2.0 anhand der Abbildungen 29, 30 und 31 als Ergebnis präsentiert. Weitere Details sind im Anhang 10.7 in der technischen Dokumentation enthalten.

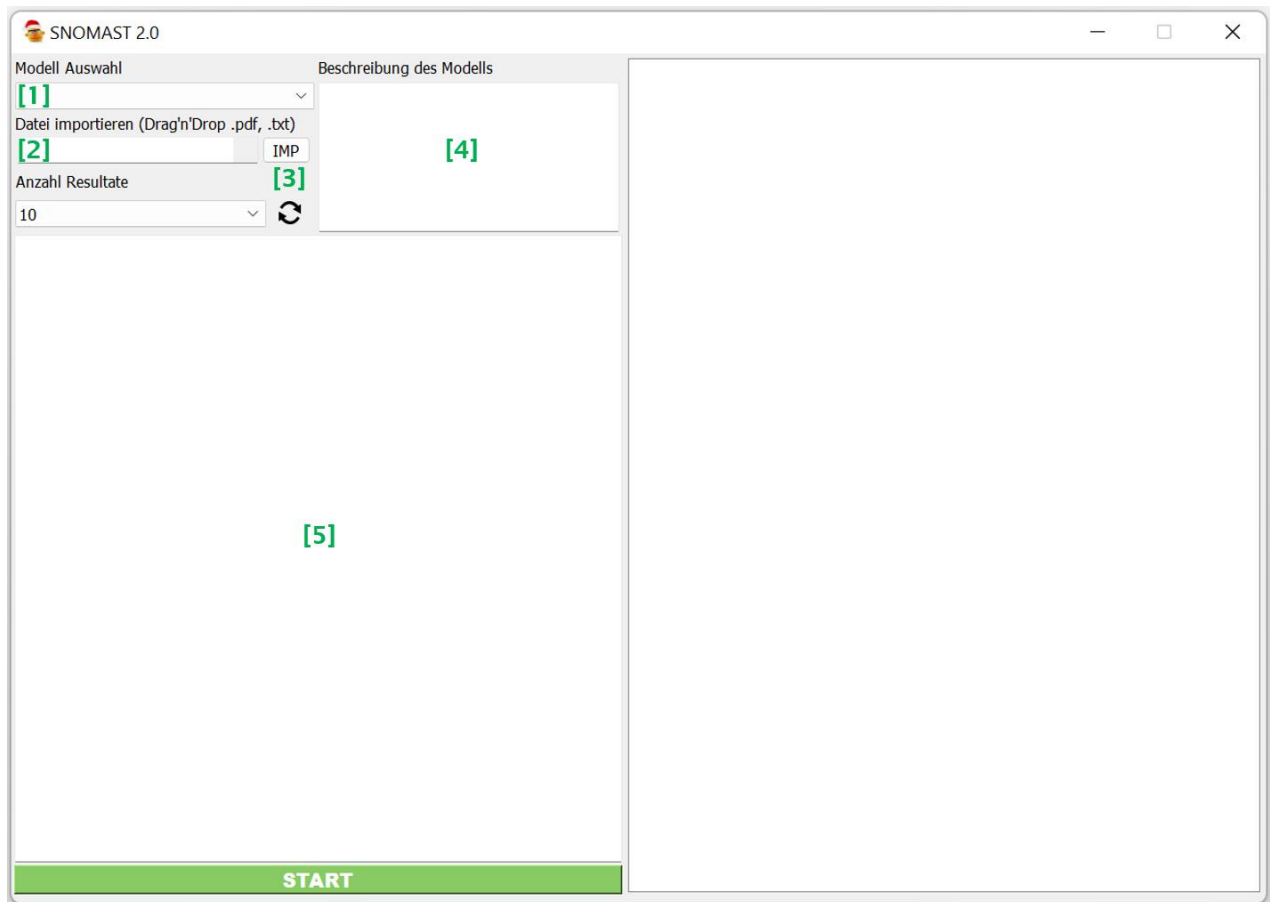


Abbildung 29: SNOMAST 2.0 GUI - Startscreen

Nr.	GUI-Komponente	Beschreibung
1	Dropdown-Liste	Über diese Dropdown-Liste kann eines der drei verfügbaren Modelle ausgewählt werden. Diese kann bei Bedarf mit neuen Modellen erweitert werden.
2	Drag & Drop Feld	Per Drag & Drop kann der Pfad zur gewünschten Datei hinterlegt werden.
3	Button	Mit Klick auf den Button «IMP» wird der Import der Datei, dessen URL im Drag & Drop Feld [2] hinterlegt ist, gestartet und dessen Text im Feld [5] angezeigt.
4	Feld	Wird ein Modell aus der Dropdown-Liste ausgewählt [1], wird in diesem Feld die Beschreibung des Modells aufgeführt.
5	Feld	In diesem Feld kann ein Text direkt über die Tastatur erfasst, via Ctrl + V eingegeben oder durch den IMP-Button [3] importiert werden.

Tabelle 17: Beschreibung GUI Komponenten - Startscreen

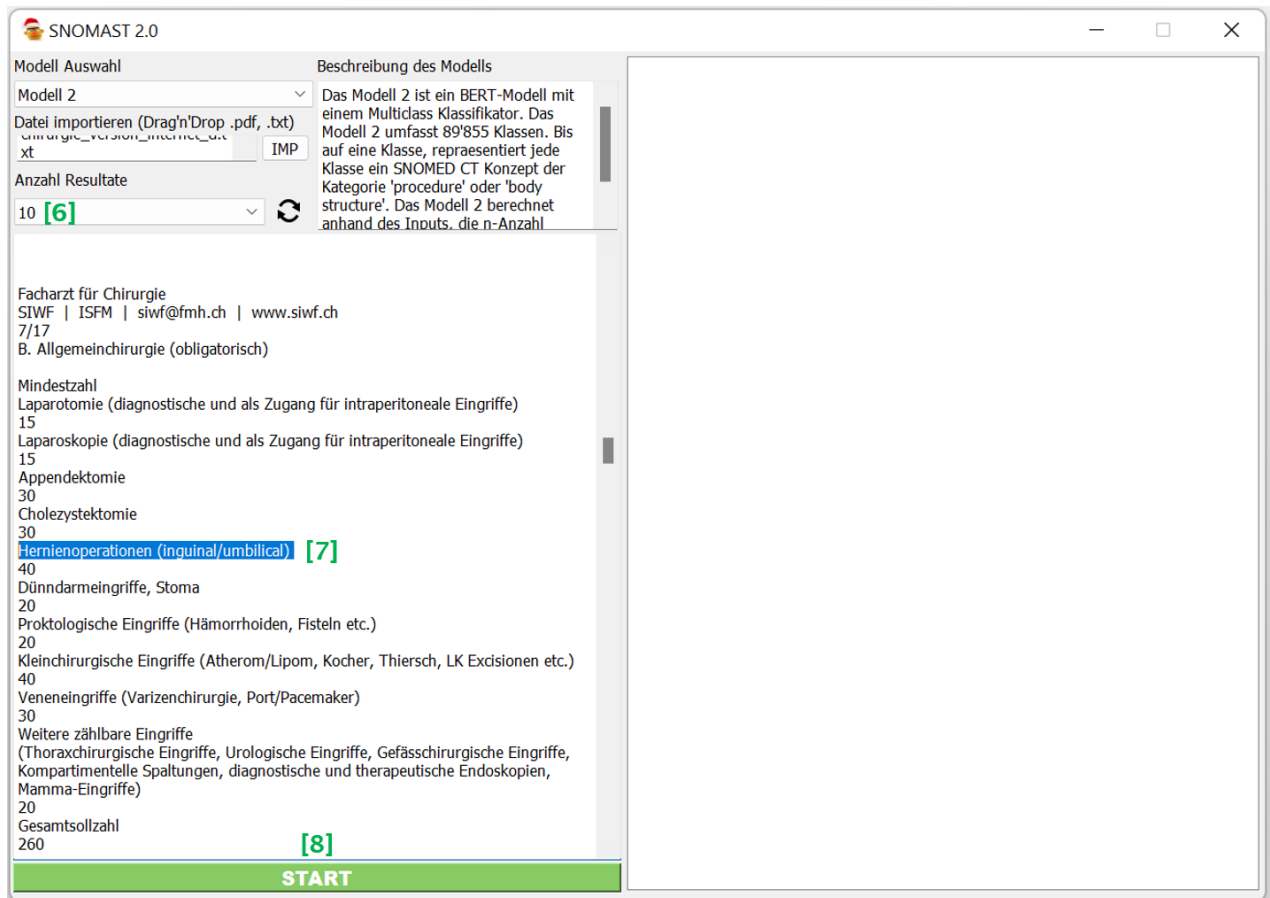


Abbildung 30: SNOMAST 2.0 GUI – Inputscreen

Nr.	GUI-Komponente	Beschreibung
6	Dropdown-Liste	Über diese Dropdown-Liste kann die Anzahl der anzuzeigenden Resultate im Output Feld (Feld rechts [12]) ausgewählt werden.
7	Textauswahl	Im Feld [5] kann durch Markieren der Text für das Entity-Linking ausgewählt werden.
8	Button	Mit Klick auf den Button «START» wird das Entity-Linking mit dem Modell [1] gestartet. Falls kein einzelner Text markiert wird, werden von SNOMAST 2.0 die ersten 3000 Zeichen verarbeitet.

Tabelle 18: Beschreibung GUI Komponenten - Inputscreen

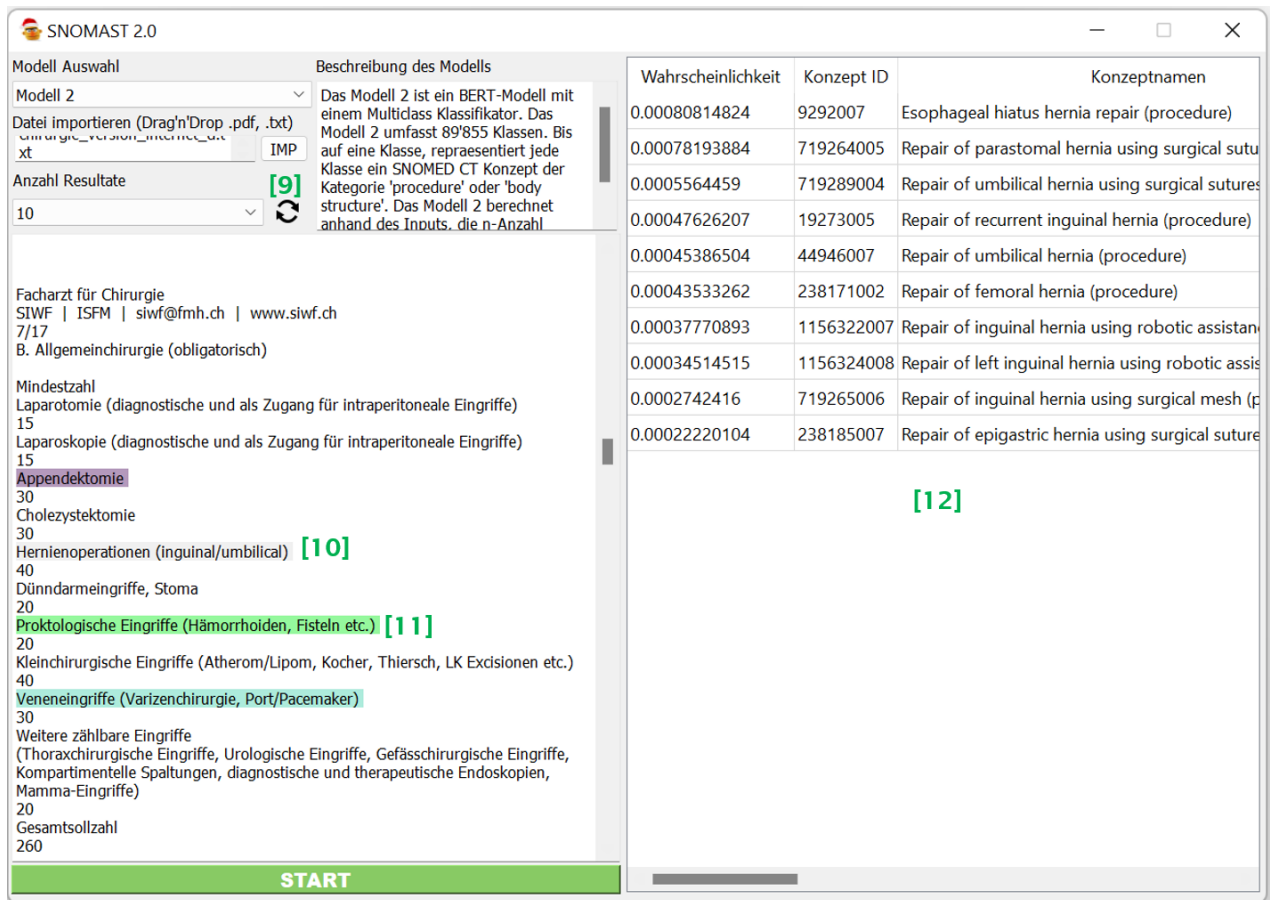


Abbildung 31: SNOMAST 2.0 GUI – Output Screen

Nr.	GUI-Komponente	Beschreibung
9	Icon Button	Dieser Icon Button «Refresh» erlaubt es, die Anzahl der angezeigten Resultate dynamisch, durch eine angepasste Auswahl in der Dropdown-Liste [6], zu aktualisieren.
10	Textmarkierung	Der ausgewählte Text wird in der Farbe grau hervorgehoben.
11	Textmarkierung	Textabschnitte, die bereits verwendet wurden, werden farblich hervorgehoben
12	Feld (Tabelle)	<p>Nach Abschluss vom Entity-Linking wird der Output in einer Tabelle mit n Ergebnissen in vier Spalten dargestellt. Anzahl Ergebnisse = Auswahl der Resultate [6]. Die vier Spalten sind:</p> <p><u>Wahrscheinlichkeit</u> Gibt in absteigender Reihenfolge die Wahrscheinlichkeit des gemappten SNOMED CT Konzeptes an.</p> <p><u>Konzept ID</u> Identifikator vom SNOMED CT Konzept (engl. Concept Unique Identifier / Cui).</p> <p><u>Konzeptname</u> Der bevorzugte (engl. preferred) Name des SNOMED CT Konzeptes.</p> <p><u>Konzept ID   Konzeptname</u> Kombination von Konzept ID und Konzeptname für ECL-Queries (ist in Abbildung nicht ersichtlich). Beispiel: 86481000 Laparotomy</p>

Tabelle 19: Beschreibung GUI Komponenten - Output Screen

#### 4.4 Bereich Gesamtevaluation

Es werden die Ergebnisse der Evaluierung der Modelle anhand des kompletten Trainingsdatensatz und der System Usability Scale – Score (SUS Score) präsentiert. Für die aufgeführten Ergebnisse der Metriken und Leistungsmerkmale wurden die gesamten Trainingsdaten (siehe Tabelle Übersicht Merkmale Modelle) betrachtet. Das heisst, jeder einzelne Text ist dem Modell als Input übergeben worden und das Resultat floss in die Gesamtevaluation.

##### 4.4.1 Leistungsmerkmale und Metriken

Der Output der Classifier gibt mittels Softmax-Funktion eine Wahrscheinlichkeit der Zugehörigkeit des Input zu einer Klasse aus. Dies führt dazu, dass jedes Mal für jede Klasse eine Wahrscheinlichkeit ausgegeben wird. Um den Output dennoch messbar machen zu können, haben wir uns entschieden, dass jeweils die Klasse mit der höchsten Wahrscheinlichkeit für die Evaluation verwendet wird.

In nachfolgender Tabelle 20 werden die Ergebnisse der Leistungsmerkmale und der Metriken der Modelle präsentiert.

Metriken	Modell 1		Modell 2		Modell 3	
	n	%	n	%	n	%
True Positive	241'200	51.93	133'736	55.40	1'770	71.57
True Negative	223'007	48.01	112	0.05	181	7.32
False Positive	123	0.03	106'887	44.28	357	14.44
False Negative	171	0.04	668	0.28	165	6.67
Total	464'501	100.00	241'403	100.00	2'473	100.00

Tabelle 20: Ergebnisse Metriken

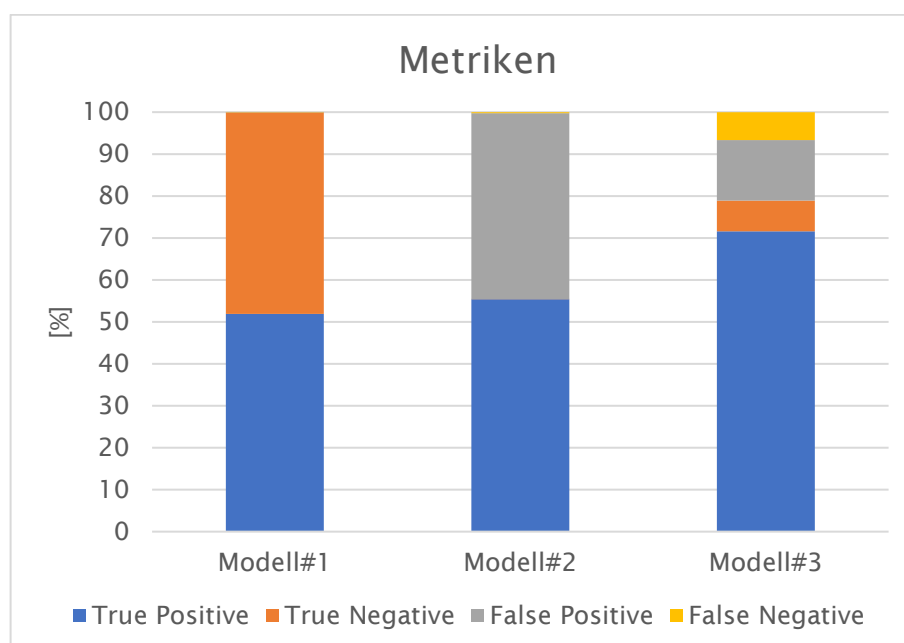


Abbildung 32: Metriken - Säulendiagramm gestapelt



Leistungsmerkmal	Modell 1	Modell 2	Modell 3
	%	%	%
Accuracy	99.94	55.45	78.89
F1	99.94	71.32	87.15
Recall	99.93	99.50	91.47
Precision	99.95	55.58	83.22

Tabelle 21: Ergebnisse Leistungsmerkmale

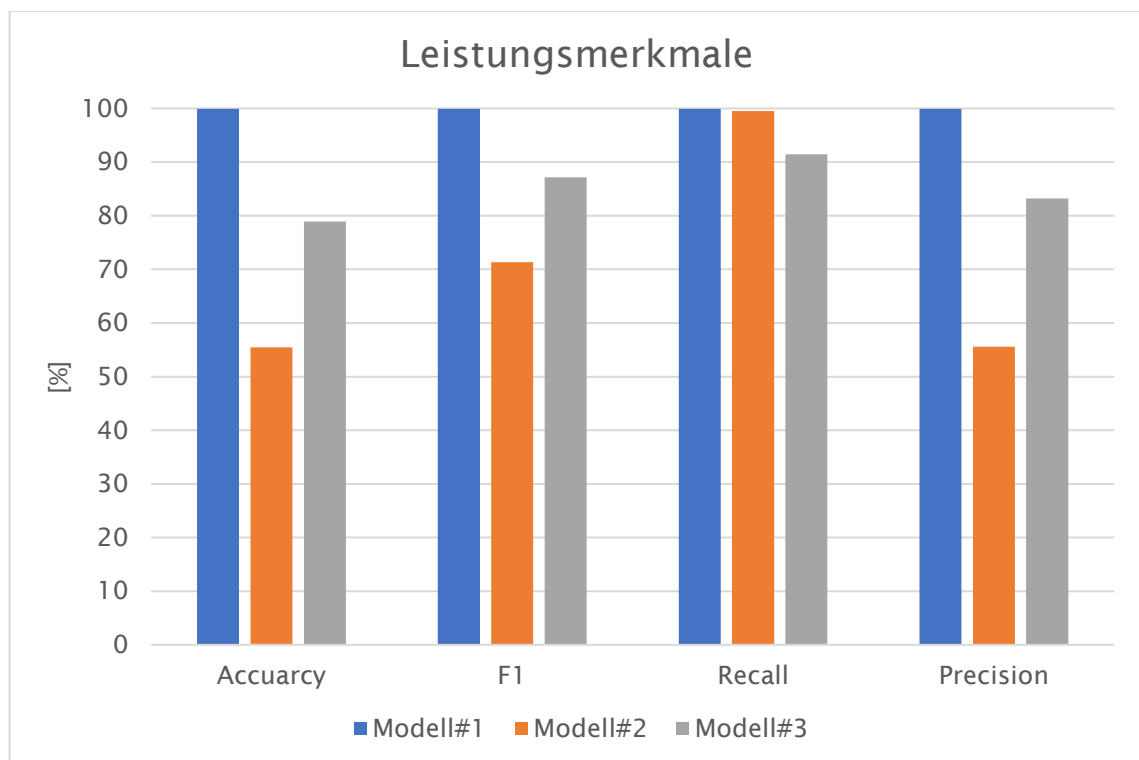


Abbildung 33: Leistungsmerkmale - Säulendiagramm gruppiert

#### 4.4.2 SUS Score

Der SUS Score ist mittels der Webplattform «SurveyMonkey» durchgeführt worden und wurde wie geplant zweimal durch das SIWF ausgefüllt. Die Ergebnisse werden in der Tabelle 22 mit der Punktzahl zusammengefasst.

Nr.	Aussage	Mögliche Antworten				
		Stimme nicht zu	Stimme eher nicht zu	Neutral	Stimme eher zu	Stimme zu
1	Ich kann mir sehr gut vorstellen, das System regelmässig zu nutzen.	-	-	-	6	-
2	Ich empfinde das System als unnötig komplex.	4	3	-	-	-
3	Ich empfinde das System als einfach zu nutzen.	-	-	-	3	4
4	Ich denke, dass ich technischen Support brauchen würde, um das System zu nutzen.	4	3	-	-	-
5	Ich finde, dass die verschiedenen Funktionen des Systems gut integriert sind.	-	-	-	3	4
6	Ich finde, dass es im System zu viele Inkonsistenzen gibt.	4	3	-	-	-
7	Ich kann mir vorstellen, dass die meisten Leute das System schnell zu beherrschen lernen.	-	-	-	-	8
8	Ich empfinde die Bedienung als sehr umständlich.	4	3	-	-	-
9	Ich habe mich bei der Nutzung des Systems sehr sicher gefühlt.	-	-	-	3	4
10	Ich musste die Hilfe der Entwickler beanspruchen, bevor ich mit dem System arbeiten konnte.	8	-	-	-	-
Summe		24	12	0	15	20
SUS Score (Gesamtsumme/2*2.5)		88.75				

Tabelle 22: Ergebnisse System Usability Scale

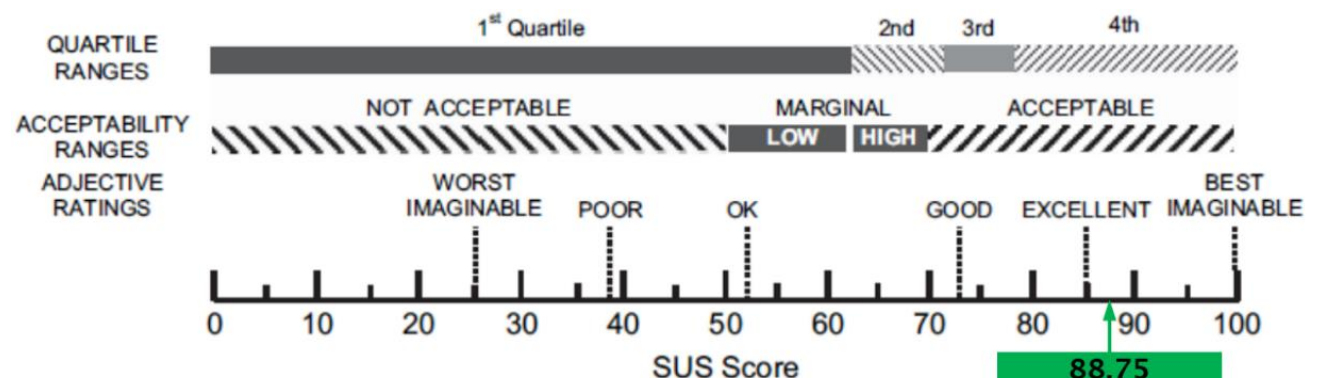


Abbildung 34: Einordnung Ergebnisse SUS Score

#### 4.5 Zusammenfassung Ergebnisse

Dieses Kapitel fasst die Ergebnisse der vorherigen Kapitel (exkl. Prototyp und GUI) für die Diskussion in der Tabelle 23 zusammen.

Ergebnis Typ	Modell 1	Modell 2	Modell 3
Trainingsdauer	24 Std.	264 Std. / 11 Tage	3.3 Std.
Grösse Trainingsdaten [Anzahl Texteinträge]	464'501	241'403	2'473
True Positive <sup>1</sup>	51.93 %	55.40 %	71.57 %
True Negative <sup>1</sup>	48.01 %	0.05 %	7.32 %
False Positive <sup>1</sup>	0.03 %	44.28 %	14.44 %
False Negative <sup>1</sup>	0.04 %	0.28 %	6.67 %
Validation F1 <sup>2</sup>	99.81 %	20.97 %	72.76 %
Test F1 <sup>2</sup>	N.A.	20.74 %	62.95 %
Evaluation F1 <sup>3</sup>	99.94 %	71.32 %	87.15 %
<sup>1</sup> basieren auf Evaluation Methode (Eval. F1), <sup>2</sup> Stichprobenraum Trainingsdaten 12.5 %, <sup>3</sup> berechnet aus kompletten Trainingsdatensatz			
SUS Score	88.75 = Excellent		

Tabelle 23: Zusammenfassung Ergebnisse

## 5 Diskussion

Zu Beginn dieser Arbeit legten wir zwei Hauptziele und fünf Fragestellungen fest. Innerhalb dieses Kapitels werden wir uns, mit Hilfe der erarbeiteten Ergebnisse, den Antworten auf die Fragestellungen annähern sowie die Zielerreichung überprüfen. Entsprechend wird dieses Kapitel im ersten Abschnitt nach den Zielen und Fragestellungen strukturiert. Im zweiten Abschnitt führen wir unsere Erkenntnisse auf und ziehen ein Fazit. Als letzter Punkt führen wir im Ausblick aus, wie unsere Arbeit weitergeführt werden könnte und was weitere mögliche Anwendungsfälle sein könnten.

Wie bereits in der Ausgangslage beschrieben, wurde in einer vorherigen Arbeit ein erster Prototyp für das Entity Linking über NLP auf Basis von MedCAT erstellt. Die mit MedCAT erstellten NLP-Modellen lieferten jedoch ernüchternde Ergebnisse (F1 Wert <0.5). Ähnlich wie bei der vorherigen Arbeit wurden auch innerhalb dieser Arbeit mehrere NLP-Modelle erstellt (Kapitel 4.2.1). Zudem hat sich während der Realisierung dieser Arbeit ergeben, dass wir drei unterschiedliche F1 Werte erhalten haben. Den Validation-, Test- und den Evaluation F1 Wert. Die NLP-Modelle sowie die F1 Werte unterscheiden sich in vielerlei Hinsicht voneinander. Nachfolgende Tabelle 24 gibt einen Überblick über diese Unterschiede.

Typ	Erläuterung
Modell 1	Das Modell 1 (Binärer Klassifikator) ist als erstes entwickelt worden. Es diente uns, im Sinne eines Proof of Concept, zur Überprüfung der Finetuning-Logik und der Schulung im Umgang mit BERT. Das Modell 1 ist für die Beantwortung der Fragestellung und Zielerreichung nicht ausschlaggebend.
Modell 2	Mit 89'855 Klassen und einer Trainingsdauer von 11 Tagen ist das Modell 2 das am aufwendigsten trainierte Modell mit den meisten SNOMED CT Konzepten. Das Potential des Modelles wurde durch das Training noch nicht komplett erreicht. Die Entwicklung des Trainings (Kapitel 4.2.3) lässt vermuten, dass mit einem längeren Training noch eine bessere Performance erreicht werden kann. Das Modell 2 entspricht der ursprünglichen Planung und Umsetzung unseres Detailkonzeptes.
Modell 3	<p>Durch die zeitliche Einschränkung der Entwicklung und dem nur langsamen Voranschreiten der Verbesserung des Modell 2, wurde von uns zeitlich etwas versetzt parallel zum Modell 2 das Modell 3 entwickelt. Hierbei handelt es sich um eine inhaltlich reduzierte Version von Modell 2. In diesem Modell wurden nur die SNOMED CT Konzepte, welche in der Parametrierung durch das SIWF verwendet wurde, einbezogen.</p> <p>Durch die Reduktion der Anzahl Klassen entstand eine Reduktion der Trainingsdaten. Dies führte wiederum zu einer Ressourcenschonung und zu einem zeitlich weniger aufwendigem Training. Ziel dieses Modelles war es herauszufinden welches Potential in einer Umsetzung des Konzeptes steckt.</p>
Validation F1	<p>Wie bereits erwähnt, wurde der Validation F1 Wert anhand des Validationsdatensatzes (Isolierte 12.5 % der Daten aus dem gesamten Trainingsdatensatz) nach jeder Epoche berechnet. Der Validation F1 dient der Einschätzung der Entwicklung eines Modelles während des Trainings auf Basis von Daten, welche während des Trainings nicht verwendet wurden.</p> <p>Da die Daten bei jeder Epoche die gleichen sind, ist die Entwicklung des Modells ersichtlich und lässt Rückschlüsse auf den Verlauf des Trainings zu. Diese Einschätzungsmöglichkeit war der Grund, weshalb wir uns für die Initialisierung vom Modell 3 entschieden haben.</p>
Test F1	Der Test F1 wird anhand des Testdatensatzes (Isolierte 12.5% der Daten aus dem gesamten Trainingsdatensatz) nach Beendigung des Trainings berechnet. Dieser F1 gibt Aufschluss darüber, wie «robust» das Modell nach Abschluss des Trainings ist.

Typ	Erläuterung
Evaluation F1	<p>Im Gegensatz zum Test und Validation F1 ist der Evaluation F1 auf Basis vom kompletten Trainingsdatensatz berechnet worden. Das führt zu einem dazu, dass wir die Überprüfung des Modells auf Basis von Daten durchführen, welche das Modell innerhalb des Trainings bereits «gesehen» hat. Zum anderen wird der F1 auf einer grösseren Datenbasis berechnet und nicht nur auf 12.5 %.</p> <p>Beim Modell 2 wird der Test F1 auf Basis von 30'175 Texteinträgen berechnet, der Evaluation F1 hingegen auf Basis von 241'403 Texteinträgen, wobei die Modelle hiervon ¼ der Einträge im Training nicht «gesehen» hatten. Diese Vorgehensweise für die Erhebung des F1 Werts ist ähnlich wie dies bereits in der vorherigen Arbeit (Living Case 2) gemacht wurde und daher am ehesten mit dieser vergleichbar.</p>

Tabelle 24: Erläuterung Unterschiede Modelle und F1 Werte

Nun lautet die Frage «Welcher F1 Wert von welchem Modell ermöglicht eine Gegenüberstellung zum erhobenen F1 Wert aus der vorhergegangenen Arbeit?»

Aufgrund der in der Tabelle 24 aufgeführten Unterschiede, erlaubt der **Evaluation F1 Wert vom Modell 2** diese Gegenüberstellung. Denn dieser ist auf Basis des kompletten Trainingsdatensatzes berechnet worden. In der vorherigen Arbeit (Living Case 2) wurde ebenfalls die Evaluation mit Daten durchgeführt, welche bereits im unüberwachten Training verwendet wurden.

### 5.1 Zielerreichung Hauptziel 1

Das erste Hauptziel (1.HZ) lautete «Ein Prototyp für das Entity-Linking auf SNOMED CT über NLP auf Basis von BERT ist am Ende der BSc. Thesis erstellt und erzielt einen F1-Wert über 70 %». Die Überprüfung des Ziels erfolgt durch die Beantwortung der zu Beginn der Arbeit aufgestellten Fragestellungen:

**Welche Massnahmen führen dazu, dass das Entity-Linking gegenüber dem vorherigen Prototyp aus der Living Case 2 Arbeit verbessert wird?** (Fragestellung 1.1F)

Wie zu Beginn des Kapitels erwähnt, wird diese Fragestellung anhand des Evaluation F1 Wertes des Modell 2 beantwortet. Der vorherige Prototyp im Living Case 2 erzielt mittels Supervised Learning via MedCATTrainer beim Entity-Linking einen F1 von 50 %. Der Evaluations F1 Wert vom Modell 2 mit 71 %, steht für einen deutlich verbesserten Prototypen. Folgende Massnahmen haben zu dieser Verbesserung geführt.

#### Contextual statt Static Word Embedding

Im Vergleich zur vorherigen Arbeit, bei welcher statisches Word Embedding angewendet wurde, basiert unser Modell auf BERT, der wiederum kontextabhängiges Word Embedding anwendet.

Die Transformers Architektur mit Attention von BERT erlaubt es, Texte differenzierter zu erfassen. Durch die bidirektionale Analyse des Textes und der Anwendung von Attention wird der Zusammenhang der einzelnen Wörter besser erfasst. Zudem bietet die Erfassung des Kontextes über den gesamten Text eine detailliertere Analyse des Inhaltes. Zur Illustration dieser Verbesserung wird untenstehender Ausschnitt aus dem Operationskatalog vom Weiterbildungsprogramm der Thoraxchirurgie aufgeführt:

	Wahrscheinlichkeit	Konzept ID	Konzeptnamen
<b>Lunge</b>	0.046880715	49795001	Total pneumonectomy (procedure)
Atypische Resektionen	0.04031384	87677003	Resection of rectum (procedure)
Anatomische Segmentresektion	0.03660549	75935006	Thoracoscopic lobectomy of lung (procedure)
<b>Lobektomie, Bilobektomie</b>	0.026035529	2407009	Excision of mediastinal tumor (procedure)
Pneumonektomie	0.024898699	173172000	Excision of segment of lung (procedure)
Erweiterte Pneumonektomie			
Manschettenresektion			

Abbildung 35: Beispiel Entity-Linking mit Contextual Word Embedding

Der vorherige Prototyp mappte die Prozedur «Lobektomie, Bilobektomie» auf das SNOMED CT Konzept «125571002 | Lobectomy» – ein False Positive. Denn die Lobektomie beschreibt den chirurgischen Eingriff irgendeines Organlappens. Hier hingegen ist – für den Menschen sofort erkennbar – die Lobektomie auf die Lunge bezogen, weshalb als korrektes SNOMED CT Konzept die thorakoskopische Lobektomie der Lunge (75935006 | Thoracoscopic lobectomy of lung (procedure)) durch unseren neuen Prototyp identifiziert wird.

### One-to-many an Stelle one-to-one

Das Entity-Linking beim vorherigen Prototyp war konzeptionell ein one-to-one Mapping. Das bedeutet, dass der Prototyp vom Living Case 2 pro Wort höchstens ein SNOMED CT Konzept erkannte und als Output anzeigte. Dies führte u.a. dazu, dass bei Prozeduren, die mehr als ein Konzept abbildeten, nur eines erkannt wurde (one-to-one) und nicht mehrere (one-to-many). Beispielsweise mappte der bisherige Prototyp die Prozedur «Hernienoperationen (inguinal/umbilical)» lediglich auf 44558001 | Repair of inguinal hernia (procedure), nicht aber auch auf 44946007 | Repair of umbilical hernia (procedure) im Gegensatz zu unseren neuen Prototypen.

B. Allgemeinchirurgie (obligatorisch)			
Laparotomie (diagnostische und als Zugang für intraperitoneale Eingriffe)			
Laparoskopie (diagnostische und als Zugang für intraperitoneale Eingriffe)			
Appendektomie			
Cholezystektomie			
Hernienoperationen (inguinal/umbilical)			
Dünndarmeringriffe, Stoma			

Wahrscheinlichkeit	Konzept ID	Konzeptnamen
0.6640853	44946007	Repair of umbilical hernia (procedure)
0.080107875	44558001	Repair of inguinal hernia (procedure)

Abbildung 36: Beispiel Entity-Linking one-to-many

Das neue one-to-many Mapping wird dank der Erweiterung mit einem Classifier erreicht. Der Classifier wendet die Softmax-Funktion auf den Output des NLP-Modells an. Dadurch wird pro definiertes Label (SNOMED CT Klasse) die Wahrscheinlichkeit ausgegeben. Mit der Umsetzung der konfigurierbaren Anzahl Resultate (siehe Abbildung 30: SNOMAST 2.0 GUI – Inputscreen) werden mind. 10 bis max. 50 SNOMED CT Konzepte pro untersuchten Text angezeigt und das Entity-Linking – durch eine höhere Anzahl an möglichen SNOMED CT Konzepten und Verwendung des Kontextes ganz generell verbessert.

Mindestzahl			
Laparotomie (diagnostische und als Zugang für intraperitoneale Eingriffe)			
15			
Laparoskopie (diagnostische und als Zugang für intraperitoneale Eingriffe)			
15			
Appendektomie			
30			
Cholezystektomie			
30			
Hernienoperationen (inguinal/umbilical)			
40			
Dünndarmeringriffe, Stoma			
20			
Proktologische Eingriffe (Hämorrhoiden, Fisteln etc.)			
20			

Wahrscheinlichkeit	Konzept ID	Konzeptnamen
0.2598892	-1	none of the trained concepts
0.15788245	80146002	Excision of appendix (procedure)
0.073846415	44946007	Repair of umbilical hernia (procedure)
0.040536057	44558001	Repair of inguinal hernia (procedure)
0.034045532	238184006	Repair of epigastric hernia (procedure)
0.02825325	38102005	Cholecystectomy (procedure)

Abbildung 37: Beispiel one-to-many Mapping mit Berücksichtigung des Kontextes

### Training

Ein weiterer Unterschied der Arbeiten liegt beim Training der NLP-Modelle und dem Labelverzeichnis. In der vorangegangenen Arbeit wurde während des überwachten Lernens eine Datenbank mit Konzepten erstellt und erweitert oder Labels mit alternativen Darstellungen angereichert. Diese Darstellungen wurden wiederum bei der Verarbeitung lediglich mit dem Aufmerksamkeitsfeld des Entity-Linking Algorithmus eins zu eins verglichen. In der Umsetzung mit BERT wird der Kontext, durch den [CLS] Token bzw. Word Embedding erfasst und mit den von Beginn an festgelegten Konzepten in Beziehung gesetzt. Dadurch wird eine Art Clustering erreicht. Somit kann Unbekanntes ohne weiteres eingeordnet werden.

Ein Nachteil dieser Methode ist die fehlende Erweiterbarkeit eines bestehenden/trainierten Modells mit neuen Konzepten. Dies hat zur Folge, dass dem Preprocessing, insbesondere dem Feature Engineering, ein grösserer Stellenwert eingeräumt werden muss.

## **Wie sieht die Gesamtarchitektur aus, damit der Prototyp einfach auf neue Bedürfnisse hin, angepasst werden kann? (Fragestellung 1.2F)**

Um diese Frage zu beantworten, müssen zwei Bereiche der Arbeit beleuchtet werden. Einerseits das Finetuning eines NLP-Modells und andererseits die Umsetzung des Prototyps selbst.

### Finetuning NLP-Modell

Beim Finetuning steht jeder Schritt, der für die Erstellung eines Modells notwendig ist, in einem eigenen Jupyter Notebook zur Verfügung und wird in der technischen Dokumentation (Anhang 10.7) beschrieben. Dies erlaubt eine isolierte Ausführung und gegebenenfalls Anpassungen an unterschiedliche Anforderungen an ein Modell. Für das Training eines Modells besteht als grundlegende Anforderung einzig, dass ein Labelverzeichnis für das Modell erstellt und ein Trainingsdatensatz mit diesem Labelverzeichnis gelabelt wurde. Die Erstellung eines Labelverzeichnisses beschränkt sich nicht alleine auf SNOMED CT. Hierfür könnten auch andere Nomenklaturen, Ontologien, Terminologien und Klassifikationen etc. verwendet werden. Bei der Erstellung des Datensatzes sieht dies ähnlich aus. Solange eine klare Einteilung der einzelnen Satz-Beispiele zu einer oder mehreren Klassen möglich ist und entsprechend annotiert wurde, kann die Entwicklung eines Modells auf die neuen Bedürfnisse hin angepasst werden.

### Umsetzung Prototyp

Bei der Umsetzung des Prototyps wurde darauf geachtet, dass die Integration von neuen Modellen simpel geschehen kann. Hierfür wurde das Configuration-File integriert, welches die Verwaltung der Modelle übernimmt. Beim aktuellen Stand des Prototyps müssen neue Modelle mit ihrem Labelverzeichnis in den Projektordner gespeichert und im Configuration-File eingetragen werden. Dank dem JSON Format ist die anzuwendende Struktur und die zu hinterlegende Information nachvollziehbar und einfach durch einen Menschen möglich. Weiter kann durch die Verwendung eines virtuellen Pagemanagers der Prototyp ganz leicht auf anderen Systemen und Computern übertragen werden. Dies wurde durch den erreichten SUS Score und die positive Rückmeldung des SIWFs gezeigt. Zudem lässt sich der Prototyp dank der gewählten Model View Presenter Softwarearchitektur modular erweitern oder adaptieren. Dadurch muss bei Anpassung nicht gleich der Prototyp von Grund auf geändert werden. Diesen Vorteil konnten wir nach der Realisierungsphase direkt nutzen und den Wunsch des SIWF nach einer Statusanzeige während der Verarbeitung direkt im GUI umsetzen, ohne die anderen Komponenten verändern zu müssen.

## **Wie kann mit Transfer Learning das gewählte BERT-Modell einen F1 Wert über 70 % erzielen? (Fragestellung 1.3F)**

Zur Erinnerung, Transfer Learning eines neuronalen Netzes bedeutet, ein bestehendes Modell auf eine spezifische Aufgabe hin anzupassen. Somit wird die Struktur des Modells mit den bestehenden Layern und gelernten Gewichten übernommen und als Ausgangspunkt für die Entwicklung eines neuen Modells verwendet. Transfer Learning setzt sich grundsätzlich aus dem Pretraining und Finetuning zusammen. Wie innerhalb der Abgrenzungen erläutert, ist für unsere Arbeit auf ein eigens vortrainiertes Modell verzichtet worden und stattdessen ein bestehendes Modell – BioBERT – übernommen und mittels Finetunings um einen Classifier zu neuen Modellen erweitert worden. Im Gegensatz zur Fragestellung 1.1 wird diese Fragestellung anhand des Test F1, respektive des Verlaufs vom Validation F1 der Modelle 2 und 3 beantwortet. Da diese Leistungsmerkmale während des Finetunings erhoben wurden und sich somit direkt auf das Transfer Learning beziehen.

In dieser Arbeit wurde das Finetuning der Modelle 2 und 3 nach demselben Konzept umgesetzt. Der Unterschied der Modelle liegt in der Verkleinerung der ursprünglich angestrebten 89'855 Klassen vom Modell 2 auf 368 Klassen beim Modell 3 und der damit verbundenen Verkleinerung des Trainingsdatensatzes von 241'403 auf 2'473 Einträgen. Bei beiden Modellen zeigt sich anhand des Trainings- und Validierungsverlusts innerhalb des Trainings eine Verringerung des Trainingseffektes. Wobei sich der Validation F1 von Epoche zu Epoche im Mittel um 0.002 Prozentpunkten beim Modell 2 und 0.007 Prozentpunkten beim Modell 3 verbessert. Obwohl keiner der am Trainingsende erhobenen Test F1 einen Wert über 70 % erzielte (Modell 2 = 20.74 %, Modell 3 = 62.95 %), schlossen wir aus dem Trend des Validation F1 Wertes, dass bei beiden Modellen durch ein weiteres Training und eine Erweiterung des Trainingsdatensatzes schlussendlich ein F1-Wert von über 70 % erzielt werden könnte.



Wir halten fest, dass wir durch die von uns ergriffenen Massnahmen und dank der Gesamtarchitektur bei der Umsetzung und dem Einsatz von Transfer Learning einen Multiclass Klassifikator mit über 89'855 SNOMED CT Konzepten auf Basis von BioBERT entwickelt haben, der bei der Evaluation über den gesamten Trainingsdatensatz, mit 71.32 %, ein F1 Wert über 70 % erzielt. Gleichzeitig ist der Test F1 Wert, der einen wissenschaftlichen Vergleich mit anderen Modellen ermöglicht, mit 20.74 % deutlich unter dem vorgegebenen Zielwert. Da innerhalb dieser Arbeit primär der bisherige Prototyp mit dem neuen Prototyp anhand des F1 Werts verglichen werden sollte, mussten wir die Modelle ähnlich evaluieren, weshalb der F1 Wert über den gesamten Trainingsdatensatz berechnet wurde.

**Damit haben wir das Hauptziel Nr. 1 «Ein Prototyp für das Entity-Linking auf SNOMED CT über NLP auf Basis von BERT ist am Ende der BSc. Thesis erstellt und erzielt einen F1 Wert über 70 %» erreicht.**

## 5.2 Zielerreichung Hauptziel 2

Das zweite Hauptziel (2.HZ) lautete «Der Prototyp erzielt bei der Evaluation nach der Abnahme einen System Usability Score von mindestens 68 beim SIWF.» Um dieses Ziel zu erreichen haben wir uns zu Beginn der Arbeit nachfolgende Fragestellungen gestellt, deren Antworten wir hier nun erläutern.

**Welche Funktionen muss der Prototyp zur Verfügung stellen, damit das SIWF bei der manuellen Annotation des Mappings zwischen textuellen Beschreibungen und SNOMED CT unterstützt werden kann?** (Fragestellung 2.1F)

Um die funktionalen und nicht-funktionalen Anforderungen zu erheben und um sicherzustellen, dass der Prototyp das SIWF im Prozess auch wirklich unterstützen wird, wurde von Beginn an ein regelmässiger Austausch mit dem SIWF gepflegt. Daraus entstanden das Pflichtenheft und das Mockup, die wiederum wesentliche Bestandteile der Umsetzung waren. Die im Pflichtenheft enthaltenen Anforderungen dienten uns als Product-Backlog und als Leitlinien für die agile Realisierungsphase mit insgesamt fünf Sprints innerhalb von zehn Wochen.

Insgesamt sind 27 Anforderungen, davon 17 funktionale und 10 nicht-funktionale, im Pflichtenheft festgelegt worden. Diese wurden nach Rücksprache mit dem Betreuer und dem SIWF weiter in 19 Muss- und 8 Kann-Kriterien eingeteilt. Innerhalb der Realisierungsphase wurden 20 Anforderungen (74 %) umgesetzt. Sieben Anforderungen davon 4 Musskriterien wurden nicht umgesetzt. Welche das sind und weshalb sie nicht realisiert wurden, wird im Fazit erläutert. Nachfolgende Abbildung 38 fasst den Stand der Umsetzung der Anforderungen zusammen.

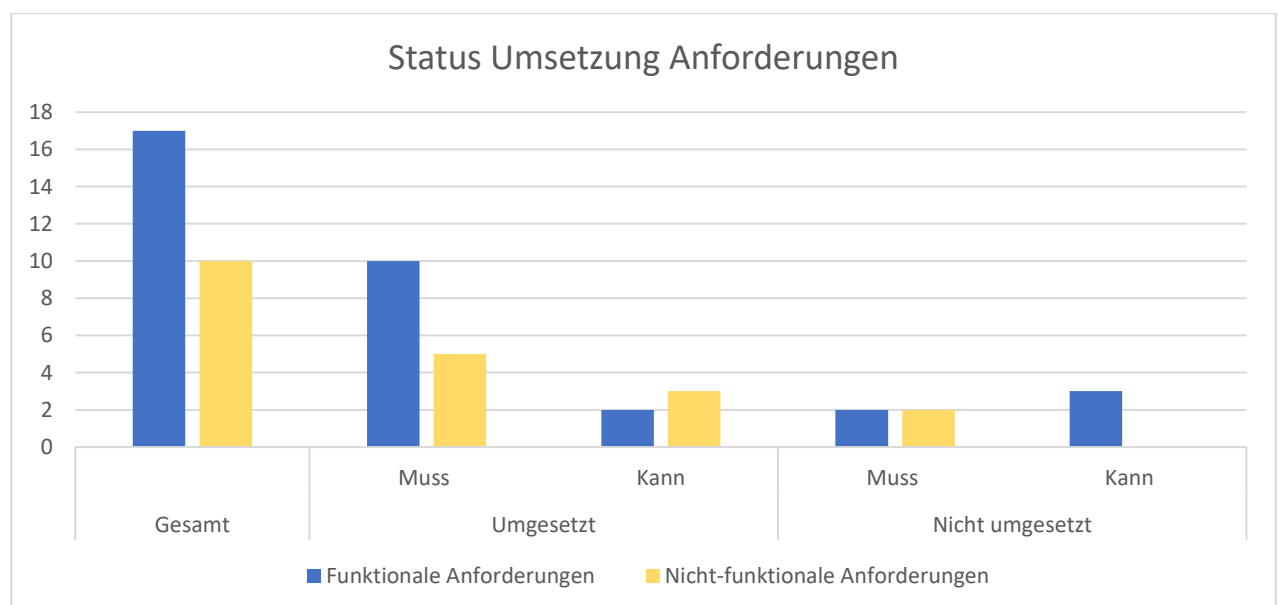


Abbildung 38: Status Umsetzung Anforderungen



Durch die im Prototyp enthaltene README Datei konnte der Prototyp ohne Probleme vom SIWF installiert, in Betrieb genommen und getestet werden. Obwohl nicht alle Anforderungen, beispielsweise BERT-Modell via einer REST-API verfügbar machen oder Visualisierung von Gütekriterien, umgesetzt wurden, können wir festhalten, dass der Prototyp die nötigen Funktionen enthält und vom SIWF eingesetzt werden kann. Diese Aussage wird gestützt durch die, in den Protokollen festgehaltenen Rückmeldungen vom SIWF und dem erreichten SUS Score von 88.75. Einziger Kritikpunkt war die fehlende Work In Progress-Anzeige, die der anwendenden Person im GUI visualisiert, dass der Text zurzeit verarbeitet wird. Dieser Punkt wurde aufgenommen und nachträglich noch im GUI implementiert.

In Bezug auf die Modelle wurde das Modell 3 positiv hervorgehoben. Bei der Nutzung mit unbekannten Texten zeigte sich eine hohe Übereinstimmung der Resultate des Modells und den manuell gemappten Konzepten. Zudem sei es hilfreich gewesen, Resultate zu sehen, an welche die parametrierende Person nicht gedacht habe.

**Was sind geeignete Visualisierungsmöglichkeiten vom Word Embedding, damit das SIWF bei der Einschätzung des Modells unterstützt wird? (Fragestellung 2.2F)**

Die Anforderungen, welche die Visualisierung von Word Embeddings zur Einschätzung der Einbettung der Resultate im Vokabular vorgaben, wurden in dieser Arbeit nicht umgesetzt. Die Umsetzung ist nach Rücksprache mit dem SIWF und unserem Betreuer innerhalb der agilen Entwicklungsphase, posteriorisiert worden. Der Entscheid der Posteriorisierung erfolgte aufgrund von Verzögerung in der Entwicklungsphase. Wie im Kapitel Erkenntnisse weiter erläutert wird, wurde der Aufwand des Preprocessings der Daten sowie die langwierigen Trainingsphase der Modelle unterschätzt. Nichtsdestotrotz wurde für die Visualisierungsmöglichkeiten vom Word Embedding eine konzeptuelle Vorarbeit geleistet und ein geeignetes Tool, namentlich TensorBoard, innerhalb der Variantenanalyse eruiert.

Zusammenfassend können wir festhalten, dass der Prototyp dem SIWF alle notwendigen Funktionalitäten für das Mapping von Prozeduren auf SNOMED CT Konzepte zur Verfügung stellt. Obschon nicht alle Anforderungen aus dem Pflichtenheft umgesetzt wurden, erhielt der Prototyp mit einem SUS Score von 88.75 eine ausgezeichnete Bewertung.

**Somit haben wir das Hauptziel Nr. 2 «Der Prototyp erzielt bei der Gesamtevaluation nach der Abnahme einen System Usability Score von mindestens 68 beim SIWF.» ebenfalls erreicht.**

### 5.3 Erkenntnisse

Unsere gewonnenen Erkenntnisse der letzten 17 Wochen werden in diesem Kapitel festgehalten und von uns reflektiert. Dabei wird eingangs die Erkenntnis in einem Satz präsentiert und im anknüpfenden Abschnitt erläutert.

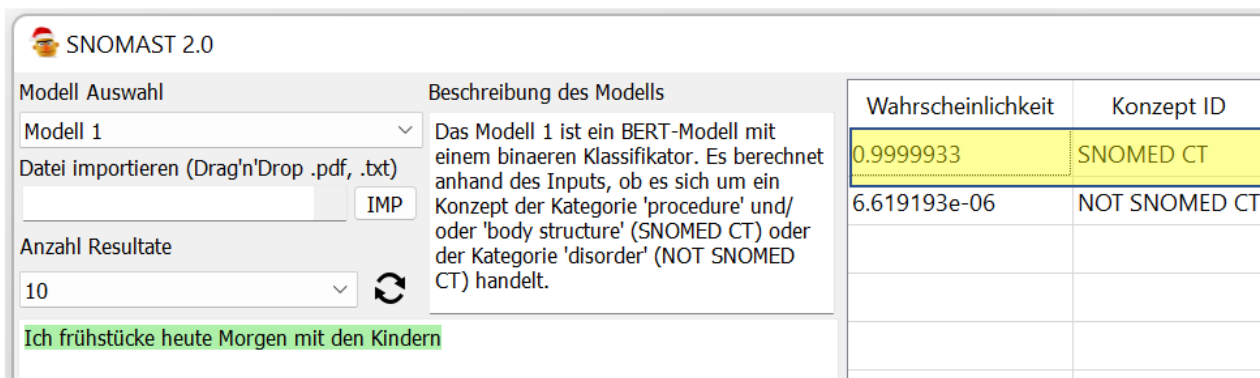
#### Einen binären Multiclass Klassifikator als Proof of Concept zu entwickeln war richtig

Das Modell 1 stellte in unserer Arbeit ein Proof of Concept dar und diente primär der Entwicklung des später verwendeten Workflows und der Sammlung der ersten Erfahrungen im Umgang mit Transfer Learning von BERT. Obwohl das Modell 1 das SIWF nicht beim Mapping unterstützen wird, konnte der entwickelte Workflow für das Finetuning, Struktur der Labelverzeichnisse und der Trainingsdaten nach dem Initialaufwand für das Modell 1 mehr oder weniger für das Finetuning der Modelle 2 und 3 verwendet werden.

#### Varianz der Trainingsdaten für einen Klassifikator vor dem Finetuning berücksichtigen

Durch die laufende Evaluation des Modells 1 erkannten wir, dass die Wahl der Trainingsdaten zu Nebeneffekten beim binären Klassifikator führen kann. Denn das Modell 1 erkennt primär nur SNOMED CT Konzepte der Kategorie «procedure» oder «body structure» als SNOMED CT, beziehungsweise «disorder» als NOT SNOMED CT.

Bei dieser Konzeption liessen wir ausser Acht, dass mit dem Entscheid, dass alle SNOMED CT Konzepte der Kategorie «disorder» mit NOT SNOMED CT gelabelt wurden, ein engeres Feld für NOT SNOMED CT abgesteckt worden ist. Dies führte dazu, dass domänenfremde Texte, die weder als Text der Kategorie «procedure», «body structure» oder «disorder» einzuordnen waren, fälschlicherweise mit einer hohen Wahrscheinlichkeit als SNOMED CT ausgegeben wurden, da diese Klasse eine grössere Reichweite im Rahmen des Clustering hatte (Abbildung 39).



SNOMAST 2.0

Modell Auswahl: Modell 1

Datei importieren (Drag'n'Drop .pdf, .txt):  IMP

Anzahl Resultate: 10

Beschreibung des Modells: Das Modell 1 ist ein BERT-Modell mit einem binären Klassifikator. Es berechnet anhand des Inputs, ob es sich um ein Konzept der Kategorie 'procedure' und/oder 'body structure' (SNOMED CT) oder der Kategorie 'disorder' (NOT SNOMED CT) handelt.

Wahrscheinlichkeit	Konzept ID
0.9999933	SNOMED CT
6.619193e-06	NOT SNOMED CT

Ich frühstücke heute Morgen mit den Kindern

Abbildung 39: Beispiel eines False Positive beim Modell 1

Um dieser Problematik entgegenzuwirken, müssten die Trainingsdaten für die Klasse NOT SNOMED CT eine viel grössere Varianz an unterschiedlichsten Texten aus unterschiedlichen Themengebiete enthalten. Diese Erkenntnis floss in die Entwicklung beim Modell 3 ein.

Der Trainingsdatensatz für die Klasse «none of the trained concepts» wurde mit mehr unterschiedlichen Beispielen angereichert im Gegensatz zu den restlichen Klassen, was durch die Abbildung 32 «Metriken» aus Kapitel 4.4.1 durch die True Negativ und False Negativ Werte des Modell 3 gut erkennbar ist. Diese Anpassung des Trainings zeigte beim Modell 3 bei domänenfremden Texten ihre Wirkung (Abbildung 40).

Wahrscheinlichkeit	Konzept ID	Konzeptnamen
0.03565117	-1	none of the trained concepts
0.017574858	55853002	Pelvic exenteration, female (procedure)
0.014164371	91381003	Pleural cavity structure (body structure)
0.013384964	713666005	Periacetabular osteotomy of pelvic bone (pro
0.013177446	71388002	Procedure (procedure)
0.012830365	77474007	Repair of acromion (procedure)

Abbildung 40: Beispiel eines True Negativ domänenfremden Textes bei Modell 3

### Für das Preprocessing genügend Zeit innerhalb eines Projekts einplanen

Eine essentielle Erkenntnis aus dieser Arbeit ist, dass der Erarbeitung der benötigten Klassen sowie der Aufbereitung des Trainingsdatensatzes zu Beginn der Projektplanung genügend Ressourcen eingeräumt werden müssen. Viele Klassen benötigen einen grösseren Trainingsdatensatz, was wiederum einen direkten Einfluss auf die Trainingszeit hat. Ein wesentlicher Schritt bei der Entwicklung eines Modells ist somit der Entscheid, wie viele und welche Art von Klassen das Modell haben soll (33).

Damit das Modell anwendungsfallgerecht trainiert und dennoch nicht zu überladen wird, benötigt es einen fachlichen Entscheid in Bezug auf die Tiefe der hierarchischen Stufe der verwendeten Nomenklatur und den Umgang mit den Beziehungen der Klassen untereinander. In unserer Umsetzung wurden alle Konzepte der hierarchischen Äste «procedure» und «body structure» mit einbezogen und die Beziehungen nicht berücksichtigt. Retrospektiv betrachtet ergab dies eine zu grosse Anzahl an Klassen (89'855), die während des Trainings entsprechend trainiert werden mussten. Eine spezifischere Selektion könnte helfen den Trainingsaufwand zu verringern (33).

Problematisch ist hierbei jedoch die Struktur von SNOMED CT. Diese ist polyhierarchisch aufgebaut, wodurch mehrere übergeordnete Knoten für ein ausgewähltes Konzept existieren können. Somit konnten wir nicht mit einer bestimmten Tiefe in der Hierarchie beginnen und zur Verringerung der Klassenanzahl eine Hierarchie Stufe nach oben gehen. Das bedingt eine individuelle Beurteilung pro Knoten. Aus dieser Problematik entstand unser Entscheid das Modell 3 auf Basis der bereits parametrisierten Konzepte vom SIWF zu trainieren. Dadurch konnten wir davon ausgehen, dass die ausgewählten Konzepte relevant für das Mapping beim SIWF sind.

Nach der Wahl der Klassen benötigt jede Klasse entsprechende Trainingsdaten. Gemäss recherchierter Literatur hängt die Anzahl benötigter Einträge pro Klasse für ein Training, von der angestrebten Präzision der Vorhersage ab. Als Daumenregel könne ein Minimum von 10-30 Einträge pro Klasse genutzt werden (33). Unsere Datensätze haben mit durchschnittlich vier Einträgen pro Klasse eine zu geringe Varianz und müssten bei einer Weiterentwicklung angepasst werden.

### Regelmässiges Testen mit kleineren Modellen

Die Erarbeitung von kleineren Test-Modellen und das regelmässig Testen dieser half bei der Entwicklung der geplanten Umsetzungen enorm. Dadurch konnte zuerst der Arbeitsablauf inkl. Evaluation für das Training erarbeitet werden, ohne bereits Stunden oder Tage zu verlieren, indem auf ein Resultat aus dem Training gewartet werden mussten. Zudem konnte durch die daraus entstandenen Modelle parallel zum effektiven Training die Entwicklung des Prototyps vorangetrieben werden.

### Visualisierung der Word Embedding wäre für die Beurteilung hilfreich gewesen

Während der Evaluationsphase der Modelle zeigte sich, dass eine Einschätzungsmöglichkeit bezüglich der Position eines Resultates im Vokabular oder der Vergleich auf Ähnlichkeit eines Inputs durch die Visualisierung der Word Embedding hilfreich gewesen wären. Durch diese Einschätzung hätte die Leistung eines Modells in Bezug auf dessen Clustering besser erkannt und evaluiert werden können. Dies wäre gerade bei den vom Modell falsch zugeordneten Konzepten (False-Positiv und False-Negativ) hilfreich gewesen, um die Qualität der Zuordnung besser einzuschätzen.

### Gutes Stakeholdermanagement fördert, obwohl alle nicht Anforderungen umgesetzt wurden, die Akzeptanz des Prototyps

Durch den von uns regelmässigen gepflegten Austausch mit unserem Betreuer und dem SIWF konnten wir flexibel auf Anpassungen reagieren. Es hat sich bewährt, dass wir einen wöchentlichen Austausch mit dem Betreuer hatten, um die zu dem Zeitpunkt akuten Herausforderungen und deren Lösungen zu besprechen. Zudem konnte durch die regelmässigen Sprintreviews in der agilen Realisierungsphase Rückmeldungen vom SIWF direkt abgeholt und für den nächsten Sprint eingeplant und umgesetzt werden.

Im Weiteren hat sich die Organisation der Arbeiten via Notizbuch im Microsoft OneNote gelohnt. Protokolle wurden direkt dort erfasst und mussten nicht umständlich via Mail den Teilnehmern verschickt werden. Zudem waren alle Informationen zum Stand der Arbeiten (Backlog, Zeiterfassung, Beschreibung der Arbeitspakete etc.) jederzeit transparent für die Stakeholder im OneNote verfügbar.

### Datenschutz und -sicherheit

Die in unserem Anwendungsfall verwendeten Daten sind nicht personenbezogen. Dies führte dazu, dass wir uns keine besonderen Gedanken in Bezug auf den Datenschutz und -sicherheit machen mussten.

Dennoch ist dieser Punkt für eine Weiterführung oder Anpassung an einen anderen Anwendungsfall relevant. Gerade im Bezug auf das Training eines Modelles, bei welchem grosse Datenmengen genutzt werden müssen. In dem von uns verwendeten Setting wurde das Training auf Server von «Fremdanbieter» durchgeführt und die Daten in einer Cloud (zwischen-)gespeichert. Bei «besonders schützenswerten Personendaten» müssten diese Dienste genauer betrachtet und gegebenenfalls das Setting geändert werden.

## 5.4 Fazit

Um unsere Schlussbetrachtung geordnet wiederzugeben, werden wir im ersten Abschnitt das Fazit zu der gewählten Vorgehensweise ziehen, dann unsere Schlussfolgerung zu den Ergebnissen präsentieren und im letzten Abschnitt ziehen wir die Quintessenz aus der Diskussion.

Nicht alle Anforderungen aus der Anforderungsanalyse, beispielsweise BERT-Modell via einer REST-API verfügbar machen oder die Visualisierung von Gütekriterien, wurden umgesetzt. Wir haben uns zu Beginn der Planung zu viel für den Prototypen vorgenommen und den Aufwand für das Preprocessing unterschätzt. Dennoch können wir schlussfolgern, dass wir durch die von uns gewählte Vorgehensweise und unserem Projektmanagement in der Lage waren, dem SIWF ein Prototyp auszuliefern, der ihre Erwartungen erfüllt.

Zudem zeigten wir anhand unserer gewählten Methodik auf, welche Schritte für das Finetuning eines BERT-Modells notwendig sind. Retrospektiv war es wichtig und richtig, dass wir anhand eines Proof of Concepts (Modell 1) zu Beginn der Realisierungsphase den Workflow an einem konkreten Modell durchspielten. Dadurch waren wir zwar von Beginn an beim Finetuning der Modelle stark gefordert, konnten aber anschliessend den Workflow auf die weiteren und grösseren Multiclass Klassifikatoren anwenden. Wir sind überzeugt, dass unsere gewählte Vorgehensweise eine Grundlage bietet um mittels Transfer Learning und mit einem grösseren und differenzierten Trainingsdatensatz, NLP-Modelle zu entwickeln, die einen Test F1 Wert über 70 % erreichen.

Unsere Arbeit zeigt hingegen gewisse Einschränkungen im Rahmen des Finetunings auf. Die Differenzierung und Grösse eines Modells beeinflusst massgeblich die Anforderung an den Trainingsdatensatz. Dies wirkt sich wiederum auf die Anforderung an die technische Hardware und den zeitlichen Aufwand für das Training aus. Der Zeitaufwand des Trainings sehen wir als weniger kritisch, da je nach technischer Infrastruktur das Training auch über Tage oder Wochen im Hintergrund gemacht werden kann. Bezüglich der Problematik mit der Hardware, kann mit der Anpassung der Batchgrösse einiges kompensiert werden, was sich wiederum auf die Trainingsdauer auswirken kann. Der grösste Aufwand sehen wir bei der Sammlung und Preprocessing der Daten für den Trainingsdatensatz. Diese müssen gegebenenfalls manuell annotiert werden, was ein grosser personeller und zeitlicher Aufwand darstellt. Um ein möglichst grossen Trainingsdatensatz zu erhalten, respektive diesen manuell zu annotieren, könnte auf Crowdsourcing und Clickworker gesetzt werden (34).

Durch unsere Gesamtevaluation der Modelle zeigten wir auf, dass die Modelle bereits gelernte und ähnliche Inhalte zu einem grossen Anteil richtig klassifizieren können. Dennoch lässt sich schlussfolgern, dass die Datenbasis für die Erreichung eines F1 Wertes von über 70 % die zentrale Rolle spielt, auch im Hinblick auf eine Generalisierung der Modelle. Die auf das Modell 3 bezogenen positiven Rückmeldungen des SIWFs zeigten auf, dass unser konzeptueller Ansatz in eine überzeugende Richtung führt. Trotz einer kleinen Sammlung von SNOMED CT Konzepten und eines zu kleinen Trainingsdatensatzes konnten wir ein Modell entwickeln, welches produktiv durch das SIWF genutzt werden kann. Dies bestärkt uns in der Annahme, dass mit BioBERT, einer spezifischeren Auswahl an Konzepten und einer Erweiterung der Trainingsdaten, ein generalistischeres Modell mit moderaten Mitteln erzeugt werden kann.

Summa summarum sind wir zufrieden mit unserer Arbeit. Wir haben die mit unseren Stakeholdern festgelegten Ziele allesamt erreicht. Gerne hätten wir die Visualisierung der Gütekriterien und diejenigen der Word Embedding noch umgesetzt. Wir haben uns aber bewusst für deren posteriorisierung entschieden, um stattdessen das Finetuning weiter voranzutreiben und abzuschliessen. Das Projekt, NLP für das Mapping von Weiterbildungsprogrammen auf SNOMED CT zu nutzen, wird vom SIWF weiterverfolgt und soll ausgebaut werden. Besser noch: Zusätzlich werden weitere Anwendungsbereich, wie beispielsweise die Verwendung von NLP im Rahmen der Leistungscodierung, geprüft.

## 5.5 Ausblick

Mit dieser Arbeit wurde eine Grundlage geschaffen, um weitere NLP-Modelle auf Basis von BERT zu entwickeln. Durch die hier und in der technischen Dokumentation beschriebenen Konzepte können andere Personen diese Arbeit weiterführen. Konkrete weiterführende Arbeiten wären:

- Visualisierung der Gütekriterien und Word Embedding mit TensorBoard Projector umsetzen. Die Idee dahinter ist, dass aufgezeigt werden kann, wie Word Embedding mit den False negative und False positiven zusammenhängen und so einen weiteren Einblick gewährt werden kann, worin die Ursache liegt, könnte (Workspace Analyse). Gerade in der Einschätzung der Resultate während der Entwicklung wäre ein Tool zur Überprüfung der Word Embeddings wichtig und müsste in einem weiteren Schritt angegangen werden.
- BERT-Modell via einer REST-API via Server (z.B. Flask) verfügbar machen.
- Neues, breiteres und generalisiertes Modell finetunen. Ein breiteres und generalisiertes Modell wäre wünschenswert und könnte mit moderatem Aufwand in einem weiteren Schritt, innerhalb der von uns entwickelten Code-Basis, umgesetzt werden.
- Vereinfachung Aufnahme neuer Modelle via GUI. Denkbar bei einer Weiterentwicklung wäre die Integration von neuen Modellen durch das GUI zu vereinfachen und zu automatisieren.
- Erweiterung SNOMAST 2.0 mit Schnittstelle zu Ontologie-Server. Die positiven Evaluationsergebnisse durch das SIWF führten dazu, dass neue Bedürfnisse geweckt wurden und auf Seiten des SIWF über eine Erweiterung nachgedacht wird. Eine der angesprochenen Erweiterungen ist eine Integration des Frontend des Ontologie-Servers (SNOMED CT Server) im Prototyp. Die Idee wäre, die Resultate der Modelle direkt als Startpunkt einer erweiterten Suche übernehmen zu können und so Applikationsbrüche zu vermeiden

Das von uns verwendete Konzept und die erfolgreiche Umsetzung eröffnet einige Anwendungsbereiche und Umsetzungsmöglichkeiten. Beispielsweise:

- Automatische Indexierung von Dokumenten während des Uploads ins EPD.
- Mapping und Parametrisierung von Fremddaten im Rahmen einer Schnittstelle.
- Unterstützungssysteme bei Parametrisierungsaufgaben auf unterschiedlichen Codesystemen. (ICD, CHOP o.ä.)
- Hilfssystem bei der Suche nach Leistungen in Verlaufseinträgen bei der Leistungscodierung.
- Extraktion medizinisch relevanter Daten aus Pathologieberichten und Arztbriefen, um Patientenprofile zu erweitern.
- Automatisierte Zusammenfassung von umfangreichen Krankheitsgeschichten.
- Suche nach passenden Patienten für klinische Studien. Beispielsweise kann ein NLP Modell mit den Ein- und Ausschlusskriterien für eine Kohortenstudie trainiert werden. Somit könnten parallel unstrukturierte Informationen abgefragt werden.
- Unterstützung bei der Auswertung von offenen Fragestellungen im Rahmen von Umfragen.

Bei der aktuellen Umsetzung des Prototyps beinhaltet die View dennoch etwas an Logik. Hierbei geht es primär um die Überprüfung der Inputs durch die Benutzerinnen und Benutzer und entsprechende Feedbacks durch das Programm. Diese Art der Umsetzung entstand mit Hinblick auf eine mögliche Serveranbindung, um grossen Datentransfer zu vermeiden. Dies könnte aber auch wie andere Hilfsmethoden, z.B. der Reader, in den Presenter ausgelagert werden.

Die Anwendung der Modelle beschränkt sich nicht auf den Prototyp. Einzige Voraussetzung für die Verwendung der trainierten Gewichte ist die Verwendung eines BertForSequenceClassification Modells von Hugging Face und die Anwendung eines Softmax Layers auf den Output. Somit lassen sich auch ganz andere Systemarchitekturen aufbauen, wie beispielsweise eine Verwendung der Modelle als Webanwendung.

Mit Natural Language Processing und dem heute verfügbaren BERT-Modell wird die maschinelle Verarbeitung unstrukturierter Daten möglich. Wir zeigten anhand des dezidierten Anwendungsfall bei SIWF auf, dass SNOMAST 2.0 beim Mapping zwischen textuellen Beschreibungen und SNOMED CT unterstützen kann. Wie oben genannt gibt es noch einige weitere Anwendungsfälle, bei welchen die Medizininformatik mittels Einsatzes von geeigneten NLP Modellen die bestehende Datenlandschaft besser nutzen und den Informationsgehalt verdichten kann.

Wir sind und bleiben neugierig, was noch kommen wird.

## 6 Abbildungsverzeichnis

Abbildung 1: Geschäftsprozess SIWF - Anwendungszeck SNOMAST 2.0.....	7
Abbildung 2: Anwendungsfeld NLP .....	11
Abbildung 3: Bereich NLP in KI .....	11
Abbildung 4: Semiotisches Dreieck (21) .....	12
Abbildung 5: Schematische Darstellung von Word Embeddings (22).....	13
Abbildung 6: Ablauf NLP mit NLP Modell und Classifier.....	14
Abbildung 7: Architektur neuronales Netz .....	14
Abbildung 8: Aktivierungsfunktion neuronales Netz (Feed-Forward-Netz).....	15
Abbildung 9: Transformer-Modell mit Encoder und Decoder .....	17
Abbildung 10: Self-Attention Beispiel (24) .....	18
Abbildung 11: Encoder Stack mit Self-Attention und neuronalen Netz .....	18
Abbildung 12: Gantt-Diagramm BSc. Thesis .....	21
Abbildung 13: Domänen BSc. Thesis .....	22
Abbildung 14: Detailansicht Domäne, Arbeiten und Bereiche .....	23
Abbildung 15: Ablaufdiagramm Konzept BSc. Thesis .....	24
Abbildung 16: Umgang >1 Konzepte pro Prozedur .....	27
Abbildung 17: Workflow Data collection.....	28
Abbildung 18: Dataset 1 für Training .....	29
Abbildung 19: Workflow Supervised Learning Modell 1 .....	30
Abbildung 20: Workflow Bereich Finetuning .....	30
Abbildung 21: Darstellung einer Confusion Matrix (27).....	31
Abbildung 22: System Usability Scale (28) .....	34
Abbildung 23: Validation F1 – Liniendiagramm .....	39
Abbildung 24: Modell 1 Train und Val loss .....	40
Abbildung 25: Modell 2 Train und Val loss .....	40
Abbildung 26: Modell 3 Train und Val loss .....	41
Abbildung 27: Systemdiagramm «Desktopapplikation lokal».....	42
Abbildung 28: Systemdiagramm «Desktopapplikation mit Server» .....	43
Abbildung 29: SNOMAST 2.0 GUI - Startscreen .....	44
Abbildung 30: SNOMAST 2.0 GUI – Inputscreen.....	45
Abbildung 31: SNOMAST 2.0 GUI – Output Screen.....	46
Abbildung 32: Metriken - Säulendiagramm gestapelt .....	47
Abbildung 33: Leistungsmerkmale - Säulendiagramm gruppiert.....	48
Abbildung 34: Einordnung Ergebnisse SUS Score .....	49
Abbildung 35: Beispiel Entity-Linking mit Contextual Word Embedding .....	52
Abbildung 36: Beispiel Entity-Linking one-to-many .....	53
Abbildung 37: Beispiel one-to-many Mapping mit Berücksichtigung des Kontextes .....	53
Abbildung 38: Status Umsetzung Anforderungen.....	55
Abbildung 39: Beispiel eines False Positive beim Modell 1 .....	57
Abbildung 40: Beispiel eines True Negativ domänenfremden Textes bei Modell 3 .....	58



## 7 Tabellenverzeichnis

Tabelle 1: Übersicht Hauptziele und Fragestellungen	9
Tabelle 2: Übersicht Meilensteine BSc. Thesis	20
Tabelle 3: Ergebnis und Massnamen aus Analyse hohes Risiko	21
Tabelle 4: Zusammenfassung Recherche	22
Tabelle 5: Kurzbeschreibung der Konzeptkomponenten	25
Tabelle 6: Übersicht Methodik Variantenanalysen	26
Tabelle 7: Beschreibung Metriken der Leistungsmerkmale	32
Tabelle 8: Beschreibung Leistungsmerkmale bei der Modellevaluation	33
Tabelle 9: Übersicht Fragen SUS Score	34
Tabelle 10: Ergebnis Variantenanalyse BERT-Modell	35
Tabelle 11: Ergebnis Variantenanalyse Tools zur Visualisierung der Gütekriterien	36
Tabelle 12: Ergebnis Variantenanalyse Server Framework	36
Tabelle 13: Übersicht erstellter Trainingsdatensätze	37
Tabelle 14: Übersicht erstellter Labelverzeichnisse	37
Tabelle 15: Übersicht Merkmale Modelle	38
Tabelle 16: Validation und Test F1	39
Tabelle 17: Beschreibung GUI Komponenten - Startscreen	44
Tabelle 18: Beschreibung GUI Komponenten - Inputscreen	45
Tabelle 19: Beschreibung GUI Komponenten - Output Screen	46
Tabelle 20: Ergebnisse Metriken	47
Tabelle 21: Ergebnisse Leistungsmerkmale	48
Tabelle 22: Ergebnisse System Usability Scale	49
Tabelle 23: Zusammenfassung Ergebnisse	50
Tabelle 24: Erläuterung Unterschiede Modelle und F1 Werte	52

## 8 Glossar

Begriff	Beschreibung
Accuracy	<p>Accuracy ist ein Leistungsmerkmal bei der Modellevaluation nach dem Training eines Classifiers. Es gibt das Verhältnis der richtigerweise als richtig erkannt und richtigerweise als falsch (gar nicht) erkannten Labels im Rahmen aller Daten des Datensatzes an.</p> <p>Und gibt die Genauigkeit der Vorhersage hinsichtlich der richtigen Klassifizierung an.</p> <p>Im Rahmen einer Confusion Matrix ist die Formel:</p> $\frac{TP + TN}{TP + FP + TN + FN}$
Attention	Attention ermöglicht dem Modell nützliche Informationen auszuwählen, aufzunehmen und für später zu speichern.
BERT	Bidirectional Encoder Representations from Transformers BERT ist ein von Google entwickeltes und vortrainiertes Modell (6), das in einem breiten Spektrum von NLP-Aufgaben die «state-of-the-art» Ergebnisse erzielt.
BioBert	Auf Basis von BERT mit PubMed Daten trainiertes Modell
Classifier	Im Kontext von Machine Learning ist ein Classifier ein Algorithmus, der dazu dient, einer Dateneingabe eine Klassenbezeichnung zuzuweisen. Also eine Art automatisierte Kategorisierung des Inputs. Ein Klassisches Beispiel ist die Spam-Erkennung, hierbei werden E-Mails in die Klasse Spam oder kein Spam eingeteilt.
ClinicalBERT	Auf Basis von BioBERT mit MIMIC Daten trainiertes Modell
Contextual Word Embeddings	Das Ziel von Contextual Word Embeddings ist die Mehrdeutigkeit eines Wortes korrekt darzustellen. Dieser Ansatz der Word Embeddings wird bei Verfahren mit Neuronalen Netzen verwendet und durch diese berechnet. Für die Berechnung benötigt es Aufmerksamkeits-Mechanismen, welche den Kontext (n-Worte vorher und nachher) berücksichtigen.
Cos-Similarity	Cos-Similarity ist ein Gütekriterium, dessen Analyse dient dem Ähnlichkeits-Vergleich von zwei Wörtern oder Sätzen im Rahmen des gegebenen Vektorraumes.
Entity Linking	<p>Entity Linking bezeichnet, im Rahmen der Verarbeitung der natürlichen Sprache, die Aufgabe einer im Text erwähnten Entität (Person, Ort, Unternehmen) eine eindeutige Identität zuzuweisen. Somit ist sie eine auf der Named Entity Recognition (NER) aufsetzende Aufgabe. NER hat die Aufgabe die Entität zu identifizieren, jedoch nicht festzustellen, um welche spezifische Entität es sich handelt.</p> <p>Named Entity Linking hat verschiedene Herausforderungen, welche gelöst werden müssen:</p> <ul style="list-style-type: none"> <li>- Variation der Repräsentation</li> <li>- Mehrdeutigkeit</li> <li>- Fehlende Abbildung in der Wissensdatenbank</li> <li>- Skalierbarkeit und Geschwindigkeit</li> <li>- Sich ändernde Informationen</li> <li>- Mehrsprachigkeit</li> </ul>

Begriff	Beschreibung
F1	<p>F1 ist ein Leistungsmerkmal bei der Modellevaluation nach dem Training eines Classifiers. Und beschreibt die Gesamtgenauigkeit der Vorhersagen.</p> <p>Dazu werden Precision und Recall gleich gewichtet.</p> $\frac{2 \times Precision \times Recall}{Precision + Recall}$ <p>Es gilt als kombiniertes Mass, dass häufig in Literaturen zu Modellen angegeben wird.</p>
Finetuning	Finetuning bedeutet, dass die Gewichte eines trainierten neuronalen Netzes als Initialisierung für ein neues Modell verwendet werden.
FMH	Foederatio Medicorum Helveticorum. Die FMH ist der Berufsverband der Schweizer Ärztinnen und Ärzte.
Google Colab	Google Colab ist ein Cloud-Dienst von Google. Er basiert auf der Jupyter Notebook Umgebung und dient der Ausbildung und Forschung im Bereich des maschinellen Lernens. Die Plattform ermöglicht es Machine-Learning-Modelle direkt in der Cloud zu trainieren und bietet hierfür speziell Ressourcenallokation vom Server an.
Hugging Face	Hugging Face ist ein auf KI spezialisiertes Unternehmen. Sie stellen eine Plattform für eine Community zur Verfügung, über welche KI-Modelle ausgetauscht werden können. Zudem entwickelten sie APIs für die Entwicklung von <i>Transformer</i> Modellen und bieten auch eine direkte Interaktion mit Google Colab an.
IHTSO	International Health Terminology Standards Development Organisation (IHTSDO) ist eine Non-Profit-Organisation, die SNOMED CT entwickelt und verbreitet.
Keras	Keras ist eine Open Source Deep Learning Programmbibliothek, geschrieben in Python. Keras bietet eine einheitliche Schnittstelle für TensorFlow und andere Bibliotheken. Das Ziel ist die Anwendung von TensorFlow, und den anderen Bibliotheken, so einsteiger- und nutzerfreundlich wie möglich zu gestalten.
Künstliche Neuronale Netze	Ein künstlich neuronales Netz versucht mit Hilfe von Algorithmen, Beziehungen in Daten zu erkennen und besteht aus n Neuronen die miteinander über mehrere Schichten (engl. Layer) verknüpft sind.
Masked-Language Modelling	Beim Masked-Language Modelling (MLM) werden zufälligerweise 15 % aller Wörter mit dem String [MASK] ersetzt und dann soll das maskierte Wort vorhergesagt werden.
Micro Frameworks	Ein Micro Framework ermöglicht in der Regel den Empfang von HTTP-Anfragen, die Weiterleitung der Anfrage an den entsprechenden Controller und die Rückgabe einer HTTP-Antwort. Dadurch sind sie speziell für APIs ohne Webanwendung/Web-App geeignet. Somit enthalten sie aber auch einen kleineren Funktionsumfang wie ein Full-Stack Framework. Es fehlen beispielsweise Funktionen wie Kontenverwaltung, Authentifizierung, Autorisierung etc. Macht sie aber auch etwas flexibler und einfacher in der Umsetzung.
MIMIC	Medical Information Mart for Intensive Care MIMIC ist eine Datenbank mit medizinischen Informationen von über 40.000 Patienten, die zwischen 2001 und 2012 auf der Intensivstation der Gesundheitseinrichtung «Beth Israel Deaconess Medical Center» behandelt wurden.

Begriff	Beschreibung
MVP Entwurfsmuster	Model View Presenter (MVP) ist ein Entwurfsmuster in der Softwareentwicklung, dass aus dem Model View Controller (MVC) hervorgegangen ist. Es beschreibt den Ansatz, das Modell (Logik) und die Ansicht (Benutzeransicht) voneinander zu trennen und über den Presenter als Schnittstelle zu verbinden. Dies fördert die Test- und Austauschbarkeit der einzelnen Komponenten.
Next Sentence Prediction	Bei der Next Sentence Prediction (NSP) wird in 50 % aller Paare von Sätzen der zweite Satz durch einen zufälligen ersetzt, damit das Modell die passende Abfolge lernt.
NLP	Natural Language Processing. NLP nutzt verschiedene Techniken, um natürliche Sprache zu erfassen und mit Hilfe von Regeln und Algorithmen maschinell zu verarbeiten.
Nomenklatur	Eine Nomenklatur ist eine Sammlung von Benennungen und Fachausdrücken aus einem bestimmten Themen- oder Anwendungsgebiet, die für bestimmte Bereiche verbindlich sind. Die Gesamtheit der in einem Fachgebiet gültigen Benennungen bildet eine Terminologie.
Ontologie	Eine Ontologie umfasst Konzepte, dessen Eigenschaften und wie diese in Beziehungen miteinander verknüpft sein können. SNOMED CT ist ein Beispiel einer medizinisch-orientierten ontologiebasierten Nomenklatur.
Overfitting	Überanpassung oder englisch overfitting bezeichnet eine bestimmte Korrektur eines Modells an einen vorgegebenen Datensatz. In der Statistik bedeutet Überanpassung eine zu starke Spezifizierung eines Modells auf bestimmte Daten und den Verlust an Generalität.
Precision	Precision ist ein Leistungsmerkmal bei der Modellevaluation nach dem Training eines Classifiers. Es gibt das Verhältnis der richtigerweise als richtig erkannten zu den richtig erkannten und falsch erkannten Labels an. Und gibt die Genauigkeit der Vorhersage hinsichtlich der falschpositiven Klassifizierungen an. Im Rahmen einer Confusion Matrix ist die Formel: $\frac{TP}{TP + FP}$
Pretraining	Beim Pretraining wird das BERT-Modell auf nicht gelabelte Daten über verschiedene Aufgaben wie Next Sentence Prediction und Masked-Language Modelling trainiert. Dadurch wird das Vokabular des Modelles aufgebaut. Die einzelnen Wörter erhalten, durch das Training, im Rahmen eines Vektors, ihre Einbettung im Gesamtkontextes des Vokabulars. Ziel ist hierbei die Darstellung der Bedeutung.
Principle Component Analysis (PCA)	Principle Component Analysis oder Hauptkomponentenanalyse ist ein mathematisches Verfahren der multivariaten Statistik. Ziel ist eine Dimensionsreduktion eines multidimensionalen Datensatzes. Dies wird durch eine Zusammenführung von den verschiedenen Variablen/Informationen in Hauptkomponente erreicht. Die dadurch künstlich erzeugten Variablen sind voneinander stochastisch unabhängig und erklären gemeinsam die Varianz des Datensatzes. Die PCA kann für die Visualisierung von hoch-dimensionalen Datensätze genutzt werden.
PyTorch	PyTorch ist eine von Facebook entwickelte Open Source Deep Learning Programmbibliothek.

Begriff	Beschreibung
Recall	<p>Recall ist ein Leistungsmerkmal bei der Modellevaluation nach dem Training eines Classifiers. Es gibt das Verhältnis der richtigerweise als richtig erkannte Labels zu den richtig erkannten und fälschlicherweise nicht erkannten Labels an. Und gibt die Genauigkeit der Vorhersage hinsichtlich der falsch negativen Klassifizierungen an.</p> <p>Im Rahmen einer Confusion Matrix ist die Formel:</p> $\frac{TP}{TP + FN}$
REST API	<p>Representational State Transfer oder REST ist ein Paradigma für eine Softwarearchitektur von verteilten Systemen. Das Ziel liegt in einer einheitlichen Schnittstelle für die Maschine zu Maschine Kommunikation im Client-Server Umfeld und fordert Zustandslosigkeit und Mehrschichtigkeit. Das heisst, dass in jeder Nachricht die nötigen Informationen vorhanden sind, um die Nachricht zu verstehen und für den Nutzer nur eine Schnittstelle angeboten wird.</p>
SIWF	<p>Schweizerisches Institut für ärztliche Weiter- und Fortbildung. Das SIWF ist ein autonomes Organ der FMH.</p>
SNOMED CT	<p>Systematized Nomenclature of Medicine Clinical Terms. SNOMED CT ist ein ontologiebasierter Terminologie Standard, entwickelt von der IHTSO.</p>
Supervised Learning	<p>Beim Supervised-Learning (deutsch: Überwachtes Lernen) werden gelabelte Daten in das Modell gegeben, um aus diesen Daten Muster zu erlernen, um später auf unbekannte Daten anzuwenden. Supervised-Learning wird u.a. für Regressionen und Klassifikationen genutzt, um Vorhersage von Wahrscheinlichkeiten oder numerischen Werten machen zu können.</p>
SUS Score	<p>Der System Usability Scale Score ist ein einfacher und technologieunabhängiger Fragebogen, um damit die Gebrauchstauglichkeit eines Systems zu bewerten. Es handelt sich um eine etablierte Methode zur quantitativen Analyse der Gebrauchstauglichkeit.</p>
System Usability Score	<p>Der System Usability Score ist eine auf zehn Fragen basierender Evaluationsmethode für Software.</p>
t-Distributed Stochastic Neighbor Embedding (t-SNE)	<p>t-SNE ist eine statistische Methode für die Visualisierung von hoch-dimensionalen Daten in einem zwei oder dreidimensionalen Raum. Hierfür wird eine Wahrscheinlichkeitsverteilung anhand der Ähnlichkeit der Datenpunkte berechnet und diese Wahrscheinlichkeitsverteilung wird auf die niedrigere Dimension angewendet.</p>
TensorFlow	<p>TensorFlow ist eine Open-Source Programmbibliothek für künstliche Intelligenz bzw. maschinelles Lernen. Es dient zur Programmierung und Manipulation von neuronalen Netzen.</p>
Tokenizer	<p>Ein Tokenizer ist ein Computerprogramm zur Zerlegung von Fliesstext in einzelne, für ein NLP Modell verarbeitbare Einheiten. Hierfür werden verschiedene Techniken verwendet. Meist wird ein Satz in einzelne Wörter aufgeteilt, diese wiederum in eine Grundform überführt und diese Grundform wird entsprechend dem Vokabular des NLP Modell mit seinem Word Embedding substituiert. Als Resultat entsteht eine für das NLP Modell verarbeitbare Liste von Word Embeddings, welche dem Input entspricht.</p>

Begriff	Beschreibung
Transfer Learning	Beim Transfer Learning werden Modelle für neue Anwendungsfälle genutzt ohne, dass die Modelle von Grund auf neu erstellt und trainiert werden müssen.
Transformer	Transformer ist eine Deep-Learning Architektur und wurde 2018 von Google veröffentlicht (23). Es handelt sich hierbei um die Grundarchitektur vieler vortrainierter Machine-Learning Modelle wie Bidirectional Encoder Representations from Transformers (BERT).
Word Embeddings	Bei Word Embeddings handelt es sich um eine mathematische Einbettung eines Wortes oder Symboles in einen Kontext. Hierfür wird für jedes Wort oder Symbol ein Vektor berechnet und zugeordnet. Dieser Vektor dient nun zur Darstellung der Bedeutung des Wortes und stellt eine Dimensionsreduktion dar. Durch diese Vektoren können nun auch Beziehungen von Wörtern berechnet werden.

## 9 Literatur

- [illegible]

[https://www.researchgate.net/publication/346614679\\_GottBERT\\_a\\_pure\\_German\\_Language\\_Model](https://www.researchgate.net/publication/346614679_GottBERT_a_pure_German_Language_Model).

17. Araci D. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models; 27.8.2019. Verfügbar unter: <http://arxiv.org/pdf/1908.10063v1>.
18. ukllfr/gottbert-base · Hugging Face; 2022 [Stand: 25.03.2022]. Verfügbar unter: <https://huggingface.co/ukllfr/gottbert-base>.
19. bert-base-german-cased · Hugging Face; 2022 [Stand: 25.03.2022]. Verfügbar unter: <https://huggingface.co/bert-base-german-cased>.
20. Alsentzer E, Murphy JR, Boag W, Weng W-H, Di Jin, Naumann T et al. Publicly Available Clinical BERT Embeddings; 6.4.2019. Verfügbar unter: <https://arxiv.org/pdf/1904.03323>.
21. Bürkle Thomas. Medizinische Dokumentation und Klassifikation: Grundwortschatz. Biel; 2019. (Unterrichtsskript).
22. Rozado D. Using Word Embeddings to Analyze how Universities Conceptualize “Diversity” in their Online Institutional Presence. Soc 2019; 56(3):256–66. Verfügbar unter: <https://link.springer.com/article/10.1007/s12115-019-00362-9>.
23. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN et al. Attention Is All You Need; 12.6.2017. Verfügbar unter: <https://arxiv.org/pdf/1706.03762>.
24. Analytics Vidhya. Transformers In NLP | State-Of-The-Art-Models; 2019 [Stand: 08.05.2022]. Verfügbar unter: [https://www.analyticsvidhya.com/blog/2019/06/understanding-transformers-nlp-state-of-the-art-models/?utm\\_source=blog&utm\\_medium=demystifying-bert-groundbreaking-nlp-framework](https://www.analyticsvidhya.com/blog/2019/06/understanding-transformers-nlp-state-of-the-art-models/?utm_source=blog&utm_medium=demystifying-bert-groundbreaking-nlp-framework).
25. Lee D, Keizer N de, Lau F, Cornet R. Literature review of SNOMED CT use. J Am Med Inform Assoc 2014; 21(e1):e11–9. doi: 10.1136/amiajnl-2013-001636.
26. Projektmanagementplan; 2022 [Stand: 15.05.2022]. Verfügbar unter: <https://www.hermes.admin.ch/de/projektmanagement/anwenden/szenarien/organisationsanpassung/ergebnisse/projektmanagementplan.html>.
27. Daniel J, James HM. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition 2020.
28. Nachrichten, Tipps & Anleitungen für Agile, Entwicklung, Atlassian-Software und Google Cloud. Quantitative Usability-Analysen mit der System Usability Scale (SUS); 2011 [Stand: 23.05.2022]. Verfügbar unter: <https://blog.seibert-media.net/blog/2011/04/11/usability-analysen-system-usability-scale-sus/>.
29. GitHub. dmis-lab/biobert: Bioinformatics'2020: BioBERT: a pre-trained biomedical language representation model for biomedical text mining; 2022 [Stand: 31.03.2022]. Verfügbar unter: <https://github.com/dmis-lab/biobert>.
30. GitHub. EmilyAlsentzer/clinicalBERT: repository for Publicly Available Clinical BERT Embeddings; 2022 [Stand: 31.03.2022]. Verfügbar unter: <https://github.com/EmilyAlsentzer/clinicalBERT>.
31. emilyalsentzer/Bio\_ClinicalBERT · Hugging Face; 2022 [Stand: 31.03.2022]. Verfügbar unter: [https://huggingface.co/emilyalsentzer/Bio\\_ClinicalBERT](https://huggingface.co/emilyalsentzer/Bio_ClinicalBERT).
32. GitHub. umassbento/ehrbert: A fine-tuned BERT using EHR notes; 2022 [Stand: 31.03.2022]. Verfügbar unter: <https://github.com/umassbento/ehrbert>.
33. Foody GM, Mathur A, Sanchez-Hernandez C, Boyd DS. Training set size requirements for the classification of a specific class. Remote Sensing of Environment 2006; 104(1):1–14. Verfügbar unter: <https://www.sciencedirect.com/science/article/pii/S0034425706001234>.
34. Zraggen CR, Kunz SB, Denecke K. Crowdsourcing for Creating a Dataset for Training a Medication Chatbot. Stud Health Technol Inform 2021; 281:1102–3. doi: 10.3233/SHTI210364.



## 10 Anhang

## 10.1 Aufgabenstellung

## 10.2 Arbeitspakete

### 10.3 Meilensteine

## 10.4 Risikoanalyse inkl. Risikomatrix

## 10.5 Pflichtenheft

## 10.6 Mockup

SNOMAST 2.0

Modell wählen

Modell #1

Anzahl Resultate

n Anzahl

Modell Beschreibung

Beschreibung des ausgewählten Modells

Datei import (Drag'n'Drop pdf, txt)

START

Wahrscheinlichkeit	Konzept ID	Konzept Term	Konzept ID   Konzept Term
--------------------	------------	--------------	---------------------------

## 10.7 Technische Dokumentation



## 10.8 Protokolle

## 10.9 Backlog

## 10.10 Scrum Präsentationen

## 10.11 Literaturrecherche Übersicht

## 10.12 Internetrecherche Übersicht

### **10.13 SD Card**

Auf der SD Card sind alle für diese Arbeit benötigten Unterlagen enthalten. Die SD Card wird separat abgegeben an das Sekretariat BFH, Höheweg 80, 2502 Biel.

#### **10.14 Selbständigkeitserklärung**