UNIVERSITÄT AUGSBURG
Universitätsbibliothek

OPEN ACCESS
OPUS AUGSBURG

Low-resource-aware natural language processing in German medical texts

Johann Frei

Angaben zur Veröffentlichung / Publication details:

Frei, Johann. 2026. "Low-resource-aware natural language processing in German medical texts." Augsburg: Universität Augsburg.

Nutzungsbedingungen / Terms of use: **CC BY-NC-ND 4.0**

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:
CC-BY-NC-ND 4.0: Creative Commons: Namensnennung - Nicht kommerziell - Keine Bearbeitung
Weitere Informationen finden Sie unter: / For more information see:
https://creativecommons.org/licenses/by-nc-nd/4.0/deed.de

Universität Augsburg
University

IT Infrastructure for Translational Medical Research

DISSERTATION

zur Erlangung des Doktorgrades
Dr. rer. nat.

Low-resource-aware Natural Language Processing in German Medical Texts

---

Johann Frei

Universität Augsburg
Fakultät für Angewandte Informatik
Lehrstuhl für IT-Infrastrukturen für die Translationale Medizinische Forschung
eingereicht am 27. Juni 2025

Universität Augsburg
University

IT Infrastructure for Translational Medical Research

Begutachtung

Erstgutachter:	Prof. Dr. Frank Kramer
Zweitgutachter:	Prof. Dr. Elisabeth André
Drittgutachter:	Prof. Dr. Carsten Eickhoff

Tag der mündlichen Prüfung

10. März 2026

Mitglieder der Prüfungskommission für die mündliche Prüfung

Prof. Dr. Frank Kramer

Prof. Dr. Elisabeth André

Prof. Dr. Carsten Eickhoff

Prof. Dr. Annemarie Friedrich

Universität Augsburg
Fakultät für Angewandte Informatik
Lehrstuhl für IT-Infrastrukturen für die Translationale Medizinische Forschung

Abstract

Processing free-form text in the medical domain is notoriously challenging, particularly when dealing with German text. Recent advancements in data-driven machine learning for Natural Language Processing (NLP), especially with transformer-based language models, have proven to be effective in processing such text. However, the success of these approaches heavily depends on access to large, high-quality datasets, which is a major hurdle in the medical field. Privacy concerns and strict access regulations impose substantial restrictions on the availability of medical data, hindering progress in this research area. Additionally, the German language has received comparatively less attention in NLP research, resulting in fewer resources and tools for processing German medical texts. From a practical perspective, the need for privacy protection calls for models that can operate locally which renders compute-efficient models particularly important. Moreover, even when medical text corpora are available, the extensive manual annotation required for training high-quality NLP models remains a significant bottleneck, and thus, further limits the development of robust medical NLP pipelines.

This dissertation investigates a suite of methodologies designed to navigate these challenges, with a focus on German medical named entity recognition as a case study. The proposed approaches explored in this thesis range from a vocabulary-driven, search-based baseline to several neural approaches, including transformer-based architectures. In addition to a per-method evaluation in a quantitative and qualitative fashion, a shared evaluation framework across all proposed methods is presented in the context of medication detection, a critical task in medical text processing. In particular, the evaluation includes external datasets to assess the robustness and generalizability of the trained models. The baseline approach, though straightforward, offers a benchmark against which the neural models can be compared, with the latter consistently yielding superior performance.

The results indicate that most neural approaches achieve decent performance, with some models attaining remarkable F1-scores, reflecting their capability to generalize fairly well across several datasets. Nevertheless, challenges persist, particularly regarding label class divergences and dataset-specific biases, which can undermine the reliability of the obtained evaluation scores and limit broader interpretability. Furthermore, the flexibility of individual approaches varies, and thus, it influences their suitability for specific downstream tasks and use cases.

The findings underscore the necessity for open and adaptable NLP methods in the medical domain, particularly for German use cases, where data constraints are often inevitable, as well as the continued efforts towards the release of novel public corpora by the research community to evaluate developed NLP methods in more robust ways. This work provides a foundation for further research into scalable, privacy-compliant NLP solutions tailored to the unique demands of medical text processing in combination with the democratization of NLP tools and methods in this domain.

Zusammenfassung

Die Verarbeitung von Freitext im medizinischen Bereich stellt eine besondere Herausforderung dar, insbesondere bei der Verarbeitung von Texten in deutscher Sprache. Jüngste Fortschritte im Bereich des datengetriebenen maschinellen Lernens für die natürliche Sprachverarbeitung (NLP), insbesondere durch transformerbasierte Sprachmodelle, haben sich als effektiv bei der Verarbeitung solcher Texte herausgestellt. Dennoch hängt der Erfolg dieser Ansätze meist vom Zugang zu großen, qualitativ hochwertigen Datensätzen ab, was im medizinischen Kontext eine erhebliche Hürde darstellt. Datenschutzbedenken und strenge Zugriffsregelungen schränken die Verfügbarkeit medizinischer Daten stark ein und erschweren die Forschungsbedingungen in diesem Bereich. Des Weiteren ist der Fokus der NLP-Forschung auf deutsche Sprachverarbeitung verglichen mit englischer NLP-Forschung vergleichsweise gering, und zeigt sich unter anderem an der geringeren Quantität und Qualität an Ressourcen und Werkzeugen für die Verarbeitung deutscher medizinischer Texte. Zudem erfordert der Schutz sensibler Daten in der Praxis die Verwendung lokal ausführbarer Modelle, wodurch recheneffiziente Ansätze besonders relevant werden. Aber auch im Falle einer Verfügbarkeit von medizinischen Textkorpora stellt der erhebliche manuelle Annotierungsaufwand, der für das Training qualitativ hochwertiger NLP-Modelle erforderlich ist, dennoch eine zusätzliche Hürde dar, die die Entwicklung robuster medizinischer NLP-Pipelines erschwert.

Diese Dissertation untersucht eine Reihe von Methoden, die entwickelt wurden, um diese Herausforderungen zu bewältigen. Ein besonderer Fokus liegt dabei auf der Erkennung medizinischer Entitäten im Deutschen. Die in dieser Arbeit vorgestellten Ansätze umfassen sowohl einen suchbasierten Basisansatz, der auf einem vordefinierten Vokabular beruht, sowie mehrere neuronale Ansätze, die transformerbasierte Architekturen einschließen. Neben einer quantitativen und qualitativen Evaluation der einzelnen Methoden wird eine vereinheitliche Auswertung im Rahmen der Erkennung von Medikamenten, eine zentrale Aufgabe der medizinischen Textverarbeitung, durchgeführt. Die Evaluation umfasst dabei auch externe Datensätze, um die Robustheit und Generalisierbarkeit der entwickelten Modelle zu messen. Der Basisansatz dient hierbei als einfacher, aber effektiver Referenzpunkt, an dem die entwickelten neuronalen Modelle gemessen werden können, die durchweg eine überlegene Leistung zeigen.

Die Ergebnisse belegen, dass die meisten neuronalen Ansätze solide Ergebnisse erzielen. Einige Modelle erreichen dabei bemerkenswerte F1-Scores, die ihre Generalisierungsfähigkeit über verschiedene Datensätze hinweg widerspiegeln. Dennoch bestehen weiterhin Herausforderungen, insbesondere im Hinblick auf Abweichungen in den Labelklassen und datensatzspezifische Verzerrungen, die die Aussagefähigkeit der erzielten Messergebnisse sowie eine weitergehende Interpretierbarkeit erschweren können. Darüber hinaus variiert die Flexibilität der einzelnen Ansätze, was ihre Anwendbarkeit für spezifische Aufgaben und Anwendungsfälle beeinflusst.

Die Ergebnisse dieser Arbeit unterstreichen die Notwendigkeit offener und anpassungsfähiger NLP-Methoden im medizinischen Bereich, insbesondere für Anwendungsfälle in deutscher Sprache, bei denen der eingeschränkte Zugang zu Daten oft unvermeidlich sind. Ebenso unterstreichen die Ergebnisse die Wichtigkeit, neue öffentliche Textkorpora zu entwickeln und zu veröffentlichen, um die Robustheit und Generalisierbarkeit von neu entwickelten NLP-Methoden besser zu evaluieren. Diese Arbeit bietet eine Grundlage für weiterführende Forschungen zu skalierbaren, datenschutzkonformen NLP-Lösungen, die den spezifischen Anforderungen der medizinischen Textverarbeitung gerecht werden können, und trägt zur Demokratisierung von NLP-Werkzeugen und -Methoden in diesem Bereich bei.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my academic supervisor, Prof. Dr. Frank Kramer, for his invaluable guidance, constant encouragement, and unwavering support throughout the entire process of this thesis. I would also like to extend my sincere thanks to Prof. Dr. Elisabeth André and Prof. Dr. Carsten Eickhoff, who served as the examiners, for their constructive feedback and valuable input. Furthermore, I would like to express my sincere gratitude to Prof. Dr. Annemarie Friedrich for serving as a member of the examination committee during the defense.

I am profoundly grateful to my former academic mentor, Dr. Ahmad Ahmadi, at LMU, who introduced me to the scientific process and inspired me to pursue an academic path leading to this dissertation. His mentorship has been a cornerstone of my journey.

I would also like to thank my colleagues and friends, especially Dominik and Florian, with whom I shared countless memorable moments, both during work and in our free time. Their advice and support in navigating nuanced aspects of my research have been invaluable.

To my family, I owe the deepest appreciation. My heartfelt thanks go to my parents, Nadja and Robert, and my brother, Ludwig, for their endless patience and unwavering support. Their belief in me has been a constant source of strength.

Lastly, I am deeply thankful to Lisa and Simone, who provided their time and effort to help me finalize this thesis and meticulously proofread the manuscript.

To everyone who has contributed to this journey, whether mentioned by name or not, your support and encouragement have meant the world to me.

Contents

Chapter 1

Introduction

1.1 Background

The transformative developments in artificial intelligence (AI) research in the last decade have been primarily achieved by the joined integration of novel machine learning (ML) approaches and the use of a large abundance of data as training resources [1, 2]. Historically, however, notable initial roots of modern AI research efforts are often dated back to the Dartmouth workshop in 1956 [3]. At this point, early approaches were severely limited by computational resources and a low quantity of available data. In the absence of strictly data-driven approaches, gradual developments focused primarily on hand-crafted and sophisticated algorithm design and commonly materialized as computer programs that were mainly governed by search-based or rule-based approaches. Often referred to as *classical AI* or *symbolic AI*, these traditional AI systems can operate on complementary lookup tables or knowledge bases in certain instances to provide formal reasoning by combining the symbolic representation paradigm and logic inference engines [4]. Such AI programs, despite their limitations, already enabled applications that have drawn the attention of the general public in certain niches, such as route planning for navigation or superhuman chess computer programs.

In the medical context, expert systems can be attributed as an instance of such symbolic AI systems that attempt to solve domain-specific challenges such as low-level tasks like vocabulary-backed medication extraction from clinical notes or side effect tracking of drug prescriptions to potential higher-level tasks such as automated diagnostics [5, 6]. Similar to general setbacks in attempts to translate AI systems into day-to-day applications, expert systems in the medical domain, albeit niche use cases exist, are frequently considered as failed approaches [7]. Yet, the commonly observed shortcomings towards applied traditional AI systems are not limited to the medical domain, as the term *AI winter* was coined to describe the general loss of confidence in these traditional AI systems and frustration about their inherent limitations [4]. While this conception of unmet expectations did not emerge from a single, monocausal factor, these AI systems often suffered from unreliable (“brittle”) results and bad generalization abilities according to M. Mitchell [8, 9]. The observed issues are direct consequences of the methodologies employed by traditional AI systems to approach a

certain problem statement, such as the use of solely hand-crafted features and rulesets, or the integration of implicit or even explicit assumptions towards the problem into the algorithm [10]. Even though such manually engineered algorithms can enable fine-grained controls over an algorithm's behavior, determining all relevant features and rulesets in an explicit fashion becomes intractable for complex problems if the goal of specifying the actual, underlying manifold of the problem needs to be achieved [11].

With the ubiquity of the internet and its growth in terms of quantity of curated content as well as user-generated or collected data, novel large-scale datasets and compute power, primarily enabled by fast graphics processing units (GPUs), caused parts of the AI research community to pivot towards developing data-driven ML techniques such as deep neural networks to find better feature extractors and classifiers via highly data-dependent models using gradient descent instead of focusing on hand-crafted or rule-based algorithms [2]. Most notably, the success of deep neural networks, in particular convolutional neural networks, is widely acknowledged by the *ImageNet moment* when deep neural networks-based models surpassed previous image classifiers on the ImageNet dataset by a substantial margin [12]. Subsequently, research breakthroughs in the field of computer vision also expanded to other research domains such as speech recognition [13] and natural language processing (NLP) [14, 15, 16, 17]. Translating these novel techniques to applied fields remains yet an ongoing, non-trivial research question.

1.2 Motivation and Challenges

With the advent of digital transformation processes and big data in the medical fields of day-to-day healthcare services as well as clinical research, the continuous shift to increasingly data-centric analysis, evidence, and decision-making unlocks a wide variety of new potentials for the entire domain but also poses new challenges [18, 19].

In general, organizational units such as clinics and hospitals accumulate patient-related data as part of the digitalized patient admission and discharge process as well as for diagnostics, documentation and accounting purposes. Therefore, the type of data may span from basic patient data such as names and birth dates, bureaucratic and accounting-related information to fine-grained medical sensor recordings, bulk imaging or verbose medication administration data. The quality of the data, however, depends on various factors, such as the quality of the data acquisition process itself or the purpose of the data collection. [18, 19, 20, 21, 22]

The data may not be stored in a uniquely identifiable manner and may require record linkage methods to reference it to an individual patient. In order to retrieve specific data, these references may only be resolved by complex network and query operations or require additional knowledge about internal processes [23]. In order to further improve the actual accessibility to the data for certain use cases like research, data integration efforts towards the aggregation of relevant medical and clinical data into central data nodes can hereby streamline the data access process. Evidently, substantial

investments have been allocated to enable such data integration processes. [24, 25, 26, 27]

However, holistic data integration processes are unable to include information from *unstructured* data sources without further processing measures. In contrast to *structured* data, unstructured data lack a well-defined data model, and therefore information cannot be retrieved in a direct, reliable manner unlike structured data sources, where the access protocols and data format are declared and information is encoded in an explicit fashion. For use case-dependent analysis tasks as well as data integration purposes in general, information extraction methods or manual approaches are necessary.

In particular, text data is a major subset of unstructured data and therefore subject to unstructured information extraction approaches like NLP. In the clinical context, the use of clinical text is of major significance because a large portion of information is solely present in natural text. For instance, Dalianis *et al.* [28] report as a characteristic of their described *Stockholm Electronic Patient Record Corpus* that around 40% of their defined data entries are present in unstructured form. Reliable and efficient NLP methods are essential tools that enable the integration and transformation of unstructured text into structured information.

Developing novel NLP methods and models remains challenging. Apart from the use of prompt-based large language model (LLM) approaches, traditional approaches to solve common NLP tasks follow the supervised learning paradigm, requiring the availability of sufficient amounts of text data (a text corpus) in combination with corresponding annotation labels for a specific NLP task for training [29, 30]. While a general rule for the minimum amount of data samples for training an NLP model cannot be determined in general, creating an appropriately sized and annotated dataset translates to substantial efforts in both data collection and data annotation.

The raw data collection for clinical research could be, for instance, conducted as part of a prospective study, in which the data collection process is part of the study design, by using data from previous studies in a retrospective manner, or by relying on pre-existing data from routine health care services, referred to as *secondary use*. The prospective data collection is ideal for obtaining text corpora tailored well to a specific NLP use case, and can also be designed to align well with legal considerations such as patients' consents to further data use. However, this collection process often remains too costly and resource-intensive in terms of manual work needed for planning and execution. Therefore, text data from retrospective or secondary use data sources can pose a less work-intensive alternative with clear quantitative advantages [21, 31, 22]. In this case, the style and context of the pre-existing text data define the limits of the usefulness of a certain NLP objective early on. In addition, the legal ramifications of using such data can be unclear and the publication of such a corpus or even of subsequently trained models is usually prohibited without further measures.

While sufficient raw text data is necessary for many NLP tasks, those that do not rely on unsupervised or self-supervised statistical models often require one or more

additional data annotation steps to be effective. The individual annotation objective is usually strictly tied to a certain NLP task [22]. For several studies the annotation process has been described as highly expensive and effortful because it commonly involves the manual input of human experts, and the work mostly constitutes a loop of tedious procedures [32, 33]. Given the availability of human annotators, semantic disagreements on the annotation assignments among the annotators are expected to occur according to the reports, especially in cases where ill-defined or no decision boundaries have been agreed on. In some instances, this may result in low inter annotator agreement scores for certain label classes [34, 33]. Measuring, identifying and reviewing these disagreements in order to decide on a new annotation rule and update the annotation guidelines in an iterative approach can counter these problems but it eventually increases the total workload. In general, the amount of work to create the annotation data can easily exceed the resource capabilities of single researchers or small research groups.

As a consequence, the acquisition of an annotated clinical dataset poses a non-trivial task that requires two factors, namely the actual opportunity to access internal clinical texts as well as the recruitment of a team of experts to annotate such data. And while even members of clinical institutions that are eligible for accessing internal data can face the obstacle of lacking sufficient support for data annotation, legal regulations do not simply allow access to these proprietary data by researchers outside such institutions, which renders the development of NLP models difficult for these researchers [35].

In order to address the accessibility issue, data sharing initiatives are important to ease the use and access to clinical datasets. If the actual text data originates from sensitive text sources that encompass elements from common *protected health information* (PHI) factors, several measures need to be taken to guarantee the protection of privacy of individual persons, e.g. de-identification or anonymization, and hence, incur substantial additional work to the initial text acquisition steps [22]. In that regard, the MIMIC-III [36] and MIMIC-IV [37] datasets are prime examples that demonstrate the feasibility of these efforts. However, in contexts where de-identification or anonymization cannot be ensured, subsequent data cannot be shared with third parties without consent. This potentially also affects NLP models that may have been developed internally on these texts. If these models are too complex to ensure the privacy of individual patients, they cannot be shared in similar ways due to the risk of adversarial attacks, such as training data extraction [38, 39, 40].

Creative solutions can partly help to solve these frequent issues of internal, non-de-identified texts by turning the focus to domain-related, medical text with similar terminology or context but without PHI-containing or otherwise sensitive data [41, 42]. While data quality and copyright issues remain concerns, the use of data from social media and other web scraping-based approaches are viable ways to improve the current state of domain-specific data availability [43]. The use of domain-related textbooks and educational material serves as another potential and less restricted source for data acquisition [44].

While several NLP research efforts on internal clinical data have been published [45, 46], their lasting impact might often be diminished in two main aspects. First, because of the use of internal, sensitive data, the developed datasets as well as their models cannot be shared publicly as already explained in the previous paragraphs. As a consequence, this poses a practical problem because other researchers can neither freely re-use the dataset and models, nor can they build upon them or further develop them to apply these research results in other areas [41]. Second, an independent validation of the presented results is not possible and the results need to be trusted in good faith because the closed nature of the proprietary data effectively prevents scientific reproducibility attempts [35]. Given that these issues contradict the central pillars of the FAIR principles [47], it highlights the need for developing and working with open data and models that do not share such inherent shortcomings.

Several larger-scale clinical or medical datasets that are accessible to external researchers have been published. In order to obtain access to most of the datasets, their authors require interested clients to sign data use agreements as a pre-condition to granting access requests eventually. Hereby, the main focus of the international research community lies on English text processing and likewise reflects the state of published, open datasets in the English language [48, 49]. In general, the access provided with signed user agreements is usually not restricted to a limited research audience and therefore, these datasets are considered open and publicly available throughout this work, although restrictions on certain aspects like commercial usage or re-sharing may be imposed.

Successes and advances in international NLP research for English texts do not trivially translate to comparable gains for non-English, especially German texts in the medical domains [48, 49]. The German medical NLP research community has lacked larger-scale datasets of clinical texts, that have been publicly available since 2021 [34]. While modern data science often emphasizes that data quality and quantity largely determine a model's accuracy and reliability for a specific NLP task, the lack of open datasets is a significant obstacle to advancing German medical NLP research. Subsequently, it directly impacts the limited availability of tools and software frameworks for German medical text processing that could be utilized by other researchers beyond the core NLP domain [50]. In contrast, several open-source or freely-accessible tools are available in English, e.g. for research purposes, however, their dedicated target tasks, that these tools have been specifically tailored to, may differ [51, 52, 53, 54, 55, 56, 57]. In this context, a popular attempt to support other languages is by utilizing the availability of multilingual models [58]. In addition to tools that support actual downstream tasks, language models pre-trained on domain-specific data from the bio-medical, scientific or clinical context are published as well [59, 60, 61, 54, 62, 63].

Even though a few German medical or bio-medical corpora have been published [64, 32, 33, 34, 65], there are well-founded reasons to push towards increasing the number of available, monolingual text corpora.

- **Differences in Text Sample Structures**: Different corpora differ in terms of actually published text samples. In some instances, complete contexts are preserved and, for example, a single sample may correspond to a coherent doctor's note that describes the entire medical history of a patient which spans across multiple paragraphs. In other corpora, a text sample may only correspond to a single sentence and therefore, a complete semantic context can not be reconstructed anymore. While sentence-wise text samples can simplify text processing, for example, by addressing engineering challenges like limited text input sizes, they may, however, exclude the dataset from certain use cases and NLP objectives. This is especially true if, for example, the global context of a clinical document needs to be considered or if information is distributed across a longer sequence of sentences. [66, 33]

- **Domain Shifts in Text Corpora**: Published medical datasets usually originate from a specific subdomain but may even originate from a non-clinical domain, e.g. the bio-medical domain. Individual corpora therefore contribute to the overall text diversity and help to better cover potential corner cases in domain-related terminology. [67, 68, 20, 69]

- **Diversity in Linguistic Styles**: Similar to the previous point, text diversity also arise from text styles and linguistic characteristics that can also reflect institution-specific quirks and habits [21, 20, 68]. Notably, local vocabulary and abbreviations can vary significantly, especially in the German language, which differs across regions such as Austria, Switzerland, and even within local sites in Germany [70, 71].

- **Corpus Sizes**: The amount of available raw text data can enable or limit its applicability for certain NLP tasks. Unlike fine-tuning a pre-trained model for a simple downstream objective where smaller corpora may suffice, tasks such as training a raw language model from scratch strongly benefit from larger, more diverse text corpora. Although corpus sizes generally vary by orders of magnitude, even smaller, novel corpora can improve coverage of edge cases, contexts, and styles, provided that a certain quality level is maintained. Custom corpora, composed of multiple smaller datasets, can enhance model performance, improve reliability, and strengthen robustness after training.

- **Labels and Annotations**: Some text corpora do not only contain plain text but also include annotation data about the text. This metadata can vary widely in structure and semantics. Common types of annotations range from simple keywords or text classifications per document to tagging text phrases with label classes or ontology references. More complex annotations, such as time-related information or dependencies between text phrases, are also possible. Novel corpora with new types of annotation data can expand the range of NLP tasks that can be addressed using only publicly available corpora.

- **Evaluation of Existing Approaches**: In ML the evaluation of trained models is

important to better estimate the behavior of the model in situations that do not just resemble the inputs from the training environment, e.g. in order to clarify the transferability of developed methods across institutions [72, 22]. Hereby, novel corpora are beneficial. Existing approaches that work on raw text data in an unsupervised or self-supervised fashion can be evaluated on novel unseen text data while avoiding potential unwanted biases or contamination by training data artifacts in the test dataset. Similarly, novel datasets with annotation data identical to annotation data of prior text corpora can be used to verify the reliability and robustness of pre-trained models on unseen data samples.

In recent years, the research branch focused on LLMs has gained popularity as a novel tool for handling a broad range of NLP tasks. LLMs emerged from the developments around neural language models with the particular focus on processing much larger quantities of data during training compared to prior approaches along with considerably increased model size in terms of the number of trainable parameters [73]. LLMs generally follow text-to-text architectures and are usually driven by the concept of autoregressive text generation, meaning that new text is sequentially sampled from the language model while the sampling is conditioned on the sequence of previous text. In particular, this conditioning text input is a central concept of LLMs, coined as LLM prompting [74]. While models that are trained on classical language modeling tasks with text continuation objectives are common, LLM-based systems may be fine-tuned to a chat-based question-answering scheme and offer subsequent chat-like interfaces for post-training use [75]. Another, yet substantial difference to classical NLP models concerns the computation resources that are needed for training and fine-tuning such models as well as for the inference stage, which is highly relevant if such a model is applied to a larger number of samples [76, 77]. This may limit their use in e.g. data integration contexts where batch processing tasks on millions of data points are plausible tasks and renders the use of LLMs highly inefficient, especially for NLP tasks that can be solved by much smaller, classical NLP models. Depending on the actual size of the model and the workload, several LLMs are so computationally expensive that they potentially cannot be trained or deployed by a single researcher or even an academic institution without effectively shifting the computational workload to remote cloud providers. This raises similar issues regarding academic competition as mentioned previously, as well as potential reproducibility insecurities if LLMs are only integrated through a cloud API endpoint and are not open-source (or open-weights) models [78, 79].

Furthermore, for privacy-relevant data processing, the primary case when dealing with medical-related data, the processing of sensitive data at remote, cloud-based model providers may not conform to common privacy laws and regulations and cannot be directly used. Legal frameworks exist that aim to enable the use of LLMs for privacy-related, sensitive data in a legally trusted and safe environment of a cloud provider, but such legal concept relies on the correct legal behavior of all related external parties, and may not enforce the protection of sensitive data by technical measures such as

local and offline inference[1].

Novel open-source LLMs are continuously being published, many of which can be deployed on local hardware with support for GPUs or neural processing units. Hereby, several techniques exist to reduce the computational complexity of LLMs. Popular techniques include Low-Rank Adaptation tricks [80] for fine-tuning, or model weight quantization for faster inference [81]. Smaller LLMs are designed to run on limited and local hardware, often utilizing a fraction of the parameters of larger LLMs, which are typically hosted on remote cloud-based systems. However, smaller models tend to be less capable in certain NLP performance aspects compared to their larger counterparts [82].

1.3 Research Objective

Several challenges in the context of medical text analysis using NLP have already been outlined as well as the need to overcome these obstacles or to mitigate their impact. In the context of this thesis, most of these challenges can be reduced into three dimensions of low-resource limits: First, the low-resource aspect in the **human workforce** dimension is particularly relevant in contexts such as data annotation, though not limited to them. Second, the low-resource aspect in the **compute** dimension constrains the use of certain models and methods due to their demanding hardware requirements, which may exceed locally available resources. Third, the low-resource **data** aspect in terms of adequate datasets and corpora is the key factor that defines the limits of essentially any NLP project in data-driven methodologies. The primary scientific objective of this thesis is to develop and evaluate a diverse set of methods that account for the multidimensional constraints posed by limited resources.

The development of a custom NLP model to tackle a certain NLP task can range from basic, universally applicable tasks to arbitrarily use case-dependent, niche tasks. For instance, generic NLP models can implement linguistic tasks like part-of-speech tagging. On the other hand, a custom-trained NLP model may be developed for a specific relation extraction task such as, to detect the correspondence between a mention of an adverse health effect and its temporal information. To address information extraction challenges shared by multiple several institutions, this thesis works towards the detection of medical entities as a surrogate task, in particular the medication detection task. Yet, no particular dedication to specific medical NLP use cases is explicitly intended in general.

The central motivation of this thesis is to highlight and improve the state of medical NLP for German texts. As already mentioned in previous sections, the state of German medical NLP differs significantly from its English counterpart, which is globally dominant and better developed in several aspects.

[1] In the US, Microsoft's Azure OpenAI service can be used in compliance with the Health Insurance Portability and Accountability Act (HIPAA) given a signed Business Associate Agreement according to https://learn.microsoft.com/en-us/azure/compliance/offerings/offering-hipaa-us. (accessed December 1st, 2024)

NLP tools are core building blocks for transforming and processing unstructured text, making them essential utilities in the medical research domain. Thus, availability and accessibility are critical for enabling such domain-specific NLP tools. Consequently, the design and development of tools that rely on proprietary components inaccessible to independent researchers are not within the scope of this thesis. This decision also excludes the use of internal datasets from local clinics as well as non-free commercial services and solutions.

As previously mentioned, aside from classical NLP algorithms being available as locally executable programs, certain NLP models and systems may only be made available as cloud API. While the reasons for this restricted access can be numerous, e.g. due to the protection of intellectual property concerning the model itself, or technical considerations regarding model size and hardware requirements, the processing of potentially privacy-related health data at an external service provider remains a legally disputed case in the medical setting. Hence, these legal considerations can only be avoided safely by approaches that retain and run the entire data processing pipeline on local hardware instances without relying on external remote endpoints. In this thesis, the research results and development are designed to comply with the premise of offline-capable NLP processing.

1.4 Thesis Outline

This thesis is structured into the following main chapters.

Fundamentals: The chapter illustrates the key concepts of the background methodology that are prior to this work but are fundamental to the proposed approaches developed in this work. It describes the core mechanisms on the tokenization strategies and the neural NLP architectures in general, as well as important elements towards their use in named entity recognition (NER) tasks.

Materials: The developed methods are built upon a variety of pre-existing software tools, pre-trained language models, text corpora and other material and data resources underlying the developed methods. The materials are presented in this chapter and additional context information is provided to deepen the conceptual understanding.

Methods: The concepts of the proposed methods are presented in this chapter. Therefore, the chapter is further segmented into four sections in which the conceptually different approaches are described individually. The sections can be further outlined as follows:

- **Entity Linking via Knowledge Base-grounded Search**: The section addresses the work around the development of search-based methods. Because the fundamental methodology differs largely from other, more modern approaches, the chapter includes several materials and explanations that are unique to such search-based methods. The approach itself primarily serves as a baseline method.

- **Named Entity Recognition using Translation and Annotation Projection**: This section introduces an initial as well as a refined approach to acquiring a corpus and its corresponding annotation resources by the means of language translation and annotation projection. The section covers the principle design of the acquisition process and the development of a basic neural NER model as well as a fine-tuned, transformer-based NER model.
- **Named Entity Recognition using LLM-generated Annotated Corpus**: This section describes the approach for obtaining a corpus using an LLM as a data augmentation tool. A final NER model is obtained by fine-tuning a pre-trained, transformer-based NER model on the synthesized training data resources.
- **Named Entity Recognition on Weakly-annotated, WikiData-defined Corpus**: The last section presents the proposed method of utilizing Wikipedia-sourced texts along with metadata information jointly used with the WikiData knowledge graph. An NER model is fine-tuned on extracted Wikipedia texts as weakly-annotated training material.

Results: This chapter presents the results of each proposed method individually. For each method, the implementation steps and parameter choices are outlined, followed by a detailed presentation of both qualitative and quantitative results. The order of presentation corresponds to the structure outlined in the Methods chapter. Since the methods and results stem from different scientific publications with varying evaluation methodologies and datasets, this chapter proceeds with a comprehensive quantitative comparison of all developed models across multiple datasets in a unified manner, and concludes with an overview of the associated public code repositories and resource references.

Discussion: The Discussion chapter is organized into three sections:

- **Principal Findings**: A summary of key insights derived from the Results chapter.
- **Method-specific Discussion**: An analysis of approach-specific findings and considerations for each proposed method.
- **General Discussion and Limitations**: A broader reflection on overarching discussion points and study limitations that apply to more than one individual method.

Conclusion: The dissertation concludes by restating the research objectives, summarizing the developed approaches, and highlighting the key findings. Major discussion points are revisited, providing a foundation for potential future work and directions for continued research.

Chapter 2

Fundamentals

2.1 NLP Terminology and Methods

2.1.1 Common Tasks in Natural Language Processing

There are numerous applications and tasks in the field of NLP. Rather than being solely a theoretical or technical engineering effort, NLP focuses on developing tools tailored to specific use cases. As a result, NLP tasks are often designed to address particular objectives, and the methods used to solve these tasks can vary. In some cases, the objectives may be narrow and tailored to a specific challenge or application, while in others, they may be broader and applicable to generic text.

A selection of NLP tasks is described within the next subsections.

Named Entity Recognition

Named Entity Recognition (NER) is a common challenge in NLP with its goal to detect, locate and classify entities of certain label classes within a text. Given a certain text, an NER model detects and outputs a set of entities in which each entity is defined as a (*start*, *end*, *label class*) tuple. The last entry *label class* provides the entity type while the (*start*, *end*) pair corresponds to the character-wise position in the text. Hereby, *start* defines the character location of the first character of the entity and *end* defines the first character location that is no longer part of the entity. The (*start*, *stop*) pair can also be called the *span* of the named entity. While this notation does suffice in many use cases, in more complex cases of disjointed entities, where entities are interrupted within the start and the end of their span, other span representations may be preferred, such as providing a set of shorter sequence spans that can collectively define such entity.

NER is a specific type of the more general sequence tagging task. While in NER, the focus lies on actual entities, sequence tagging is much more generic in its definition and encompasses any task that assigns a certain label or *tag* element to each sequence item. The term *token classification* is closely related to sequence tagging but explicitly imposes the token-centric semantic constraint.

In generic use cases, common NER models are trained to detect basic entities such as

persons, locations or organizations, as shown in Figure 2.1. This is because popular NER training data, such as the CoNLL-2003 dataset [83], cover mostly these three entity classes. Of course, when heading towards actual use cases, e.g. in the medical domain, specialized datasets are required that convey certain label classes of interests.

Figure 2.1: Visualization of a document processed by a generic NER model.

Entity Linking

Entity Linking (EL) is a task that takes NER further to the next logical step and connects the NLP realm to the knowledge representation domain. While NER is limited to the detection of entity types with their corresponding spans within a text, EL also resolves and links the detected entities to the related entry of a certain terminology or ontology. This task may be tackled by different methodologies and encompass several subtasks to eventually achieve the goal of the initial EL task.

Relation Extraction

The Relation Extraction (RE) task aims to detect and classify relationships between named entities found in a text. These relationships may represent a variety of potential associations, such as a directed "employee of" (Person→Org) relation or a symmetric "married" (Person↔Person) relation in the context of generic NER. Other, potentially domain-specific relations may also be relevant depending on the specific use case.

In all cases, RE assumes that named entities have already been detected and thus relies on a pre-existing, well-performing NER model. Although RE is beyond the scope of this work, it is mentioned here to emphasize the importance of NER as a fundamental building block in many NLP pipelines.

Machine Translation

In machine translation, a bi-lingual or multilingual machine translation system is designed to translate a natural text sequence from a certain source language into

semantically equivalent natural text in a target language. Among numerous tasks in NLP, machine translation can be considered rather complex and poses several technical and linguistic challenges. Yet its obvious application use cases historically attracted efforts to demonstrate the feasibility and improve the quality of different machine translation approaches. Earlier approaches have used basic rule-based or statistical modeling to generate translation capabilities. In recent years, neural machine translation has been demonstrated to generally outperform previous approaches in most contexts and therefore have subsequently replaced most statistical approaches [84].

In general, NLP tasks range from basic ones like text classification and sentiment analysis to more complex tasks such as machine translation, question answering, and natural text generation. While the tasks discussed in the previous sections are most relevant to this work, other tasks, including highly interdisciplinary ones, play a significant role in addressing broader use cases within the NLP context.

2.1.2 Tokens and Tokenization Strategies

Architectures in scientific ML are often designed to process fixed-size input and output vectors. This works well for data like medical images, which typically have fixed dimensions, a consistent number of color channels, and pixel values discretized into floating-point or integer values. However, natural language is fundamentally different due to its variable structure and sequence length. Representing text as sequences of vectors introduces challenges for models with fixed-size architectures. Therefore, processing such sequences of vectors adequately requires changes to these existing architectures.

At the very technical level, text data is usually encoded as character-wise byte sequences. This simple representation suffices for data transmission and basic text processing, but it does not match the linguistic structure of natural language which rather resembles a symbol-like and thus discrete information pattern. It differs conceptually from e.g. image data that reside in the continuous domain and tends to be semantically more robust to minor corruptions in the data. In contrast, small modifications in the text sequences are highly likely to result in invalid data, or can fundamentally alter the meaning of a sentence in the text sequence.

In order to better integrate the text as a character sequence into the symbol paradigm and make efficient use of this input signal, the character sequence is often parsed and split into discrete units, called *tokens* through the process of *tokenization*. After applying a tokenizer to a character sequence, the obtained result consists of a sequence of tokens, that compose the input text as discrete units.

In a follow-up step, often referred to as token encoding, the tokens are mapped to individual token identifiers (*token IDs*), represented as integers. These identifiers can be resolved into text chunks by looking up the respective token identifier entry in the *vocabulary* table of the tokenizer. The vocabulary contains all known tokens with their corresponding token identifier. While token encoding is conceptually distinct from

the initial tokenization step, it is often implicitly treated as a substep of the broader tokenization process.

Regarding terminology, token merely refers to a lexical unit in computational linguistics, and while this is frequently associated with tokens as strings that match individual words and punctuations, no definition universally applies. The actual token instantiation depends on the selected tokenization strategy.

Different tokenization strategies exist. However, in this context, it should be considered that some mechanisms are shared between different concepts.

Rule-based Tokenization

The defining feature of rule-based tokenizers is that their tokenization strategy relies on predefined, fixed rules that are not automatically learned from the structure of specific corpora. These rules can be generic, such as splitting text by whitespace or punctuation, or they can be extended to account for language-specific linguistic characteristics, such as handling contractions in English.

Most NLP frameworks, such as Spacy, implement rule-based tokenizers, that mainly follow the paradigm of whitespace-separated tokenization with additional handling of punctuation. In this instance, the tokens mostly resemble actual words and punctuation characters [85].

Because the final tokens are mostly common words, this tokenization strategy poses an obvious choice, however in cases that assume a fixed set of vocabulary, and hence, a fixed set of tokens, this strategy may lead to troublesome edge cases. For instance, text frequently consists of compound words that can be composed arbitrarily and results in a new token entry in the vocabulary for every new compound instantiation. Here, the tokenization strategy may fail to determine a useful representation of tokens upon fixed-size vocabularies. A common solution to indicate tokenization issues in fixed-size vocabularies, is that unsupported tokens are represented by a special *out-of-vocabulary* token in the final tokenization sequence. In these cases, parts of the initial information signal are lost.

Data-dependent, Statistical Tokenization

In contrast to rule-based tokenization strategies, statistical approaches offer an alternative to the previous, rule-based approaches. The underlying concept borrows ideas from information theory and compression. It aims to find a tokenization strategy that optimizes the entropy of the probability distribution over all available tokens given a sufficiently large corpus. Due to that data-dependent strategy search, it is often referred to as a trainable tokenizer. Modern ML-based models employ tokenizers with fixed-size vocabularies (e.g. vocabulary size of 32,000 entries for Llama-based models [86, 87]), because it fits naturally to fixed-size tensor inputs of neural networks such as one-hot encodings or embedding tables that are closely linked to the weights and architecture of a model.

The principal method here is as follows: Given a sufficiently large corpus that is representative for capturing the most relevant terms in their actual occurrence frequency, an initial pre-tokenization is applied to split the text into words. These pre-tokenization splits are commonly whitespace- and punctuation-oriented since this step is rather a preprocessing step than the actual tokenization and only serves as a way to obtain a list of words to avoid working on one large character sequence. The actual, underlying rules for the pre-tokenization splits can vary and depend on individual considerations. While plain whitespace separations are easy to handle, more complex, rule-based splits can be used as well.

The words are further split into characters to determine the set of all relevant characters in the corpus and stored as initial tokens in an intermediate vocabulary, which ensures that the corpus can be later at least tokenized using the character-only tokens. Alternatively, instead of extracting all unique characters from the corpus, it is also possible to initialize the vocabulary with a fixed set of characters, e.g. ASCII symbols. To handle cases in which a word cannot be tokenized by such a vocabulary, the special out-of-vocabulary *unk* token is reserved to indicate that the word is unknown to the vocabulary.

In iterative cycles, the words are tokenized given the intermediate vocabulary. As long as this vocabulary has not reached its intended size, a token pair is selected if it applies to a certain metric, and the text of the joined token pair is added to the intermediate vocabulary as a new token entry. As a popular metric, a token pair that appears most frequently in the text corpus, is selected, merged, and subsequently reappears as a single token in the next iteration cycle. The intermediate vocabulary from the last iteration cycle is kept as the final vocabulary. Because this vocabulary consists of entries that are not words exclusively, but also instances of textual prefix and suffix patterns or other frequent artifacts, the concept is called *subword tokenization*.

Most notable implementations of such statistical tokenizations that are constructed by the training on a text corpus are as follows.

- **Byte-pair Encoding (BPE)**: The algorithm employs the initial tokenization step and determines the merge of a token pair by the highest occurrence of a pair during the iteration stage. [88]
- **WordPiece**: The WordPiece algorithm is closely related to the BPE algorithm. However, the most distinct difference from BPE is the choice of another token pair selection metric. While the BPE metric equates to picking the joined token pair with the highest frequency in the given corpus, WordPiece picks the pair such that the selection effectively increases the likelihood of the corpus after the token pair is added as a new token to the vocabulary and takes the occurrence of the pair in relation to the occurrences of the individual tokens into account. [89]
- **SentencePiece**: While both BPE and WordPiece are rather distinct concepts, SentencePiece is primarily an open-source implementation that resembles the key concepts like WordPiece and BPE and is augmented by additional, optional

features. However, it can also be perceived as a third and more flexible concept because it can be used to approach the subword tokenization problem without the use of the initial, rule-based tokenization stage. It directly works with character-based text sequences by treating whitespaces just like any other input characters. In SentencePiece, a whitespace character is represented by the ▁ underscore character. [90]

In general, tokenization strategies like rule-based or subword-oriented are also either *destructive* or *non-destructive*, referring to their ability to fully reconstruct the original input character sequence from its generated token sequence. While this property is highly dependent on the specific implementation, vanilla BPE or WordPiece tokenizers are instances that are destructive due to their initial tokenization to split text sequences into words, and therefore are unable to reconstruct sequences of multiple whitespaces. In contrast, the Spacy tokenizer is designed to guarantee a non-destructive tokenization. The term *reversible* instead of *non-destructive* tokenization is used occasionally but remains semantically equivalent.

While tokenization is a fundamental process in NLP and computational linguistics, other attempts investigate whether a text sequence can also be processed character-wise to avoid complex or inadequate tokenization effects [91]. While strictly speaking it does not entirely replace the tokenization step since the text is tokenized into individual characters, the primary goal of the tokenization to encode text more efficiently is ignored.

2.2 Neural NLP Architectures

Artificial neural networks have become primary tools for addressing and resolving NLP challenges, and gained popularity in competing with classical graphical approaches such as conditional random fields, or other statistical modeling approaches. The versatility and ability of neural networks to learn complex patterns and representations have contributed significantly to the advancement of NLP research and applications.

The core concepts that paved the way for the successes of modern language models are outlined in the following sections.

2.2.1 Attention Mechanism

Handling arbitrarily long sequences in neural networks usually requires a specific network architecture. A natural fit to these requirements is the use of recurrent neural networks (RNNs) such as Long Short-Term Memory (LSTMs) [92] network architectures. However, these approaches can exhibit major limitations in practice, such as issues with vanishing gradients where gradient magnitudes diminish over time, hindering effective learning during backpropagation (particularly in vanilla RNNs), or their inability to reliably capture long-range dependencies within the input sequence [93, 94]. Dense networks can, in theory, propagate information between input items with long-range

dependencies. However, this approach is inefficient in practice, as every connection requires implicit training and the model lacks robustness to shifts in sequence ordering. Convolutional networks, that are translation-equivariant, can be used in certain use cases of NLP but are by design not focused on capturing long-range dependencies since their receptive field is limited to local, neighboring inputs within a sequence.

Instead of attempting to process and aggregate input information solely through dense networks or sequentially via LSTM recurrent units, the concept of *attention* is introduced to selectively focus on specific parts of the input. One instance of attention is known as *hard attention*, where the neural network makes discrete decisions on which parts of the input to prioritize. Although hard attention involves non-differentiable decision-making within the neural network architecture due to its discrete nature, it can be effectively utilized across various learning regimes, including reinforcement learning. In order to jointly integrate and train an attention model via backpropagation, the non-differential aspect can be removed, and the discrete decision-making component is substituted by a differentiable *soft* implementation. It allows the end-to-end training of a neural network in which its integrated attention mechanism is fully covered by the gradient flow of the backpropagation algorithm. This *soft attention* concept is tightly linked to the *Softmax* function that forms the foundation of the soft attention mechanism.

Softmax

The Softmax function $\mathbb{R}^n \rightarrow (0,1)^n$ is defined element-wise as

$$softmax(x)_i = \frac{\exp(x_i)}{\sum_{x_j \in x} \exp(x_j)}$$

where $x \in \mathbb{R}^n$ is a real-valued vector, $i \in \{1, 2, \ldots, n\}$ is the index of the scalar in x and exp is the natural exponential function. According to this definition, the Softmax function normalizes the scalars in x into a value range between 0 and 1, ensuring that the sum over these normalized scalars sums up to 1 by normalizing the exponential scalar by the sum over all exponential scalars. It assigns relatively larger values to larger inputs in x, amplifying their relative weights compared to smaller inputs. This property causes the Softmax function to emphasize the largest value in x while still being differentiable. Therefore, the Softmax function can be interpreted as an approximation of the non-differentiable $argmax(x) : \mathbb{R}^n \rightarrow \{0,1\}^n$ function in a differentiable way. Due to these properties, the Softmax function is often used to normalize, transform, and interpret an input vector into a vector of probability scores that assign the highest probability score to the largest value of the input vector.

Attention Vectors

The soft attention concept can be defined given a sequence of context vectors $b_i \in \mathbb{R}^{d_b}$ of length l_b which contains the vectors b_i for $i = 1, 2, ..., l_b$ that can be attended to, and a second sequence of input vectors $a_i \in \mathbb{R}^{d_a}$ of length l_a which contains the vectors a_i for

$i = 1, 2, ..., l_a$ that attends to the context vectors b_i. For simplicity, it is assumed that the dimensions of the input and context vectors are the same and this shared dimension length is denoted by d_c such that $d_c = d_a = d_b$.

Within the neural network structure, three dense layers are used to predict subsequent vectors. Therefore, the matrices $W_Q \in \mathbb{R}^{d_q \times d_c}$ (**query** weight matrix), $W_K \in \mathbb{R}^{d_k \times d_c}$ (**key** weight matrix) and $W_V \in \mathbb{R}^{d_v \times d_c}$ (**value** weight matrix) are learnable parameters of the attention mechanism.

For each context vector b_i from the sequence, a query vector $q_v \in \mathbb{R}^{d_q}$ is predicted by

$$q_{b_i} = W_Q b_i$$

and can be stacked to form the query matrix $Q \in \mathbb{R}^{l_b \times d_q}$ to assemble the individual query vectors for each sequence item.

Similarly, for each input vector a_i, the key and value vectors are predicted by a matrix-vector multiplication.

$$k_{a_i} = W_K a_i$$

$$v_{a_i} = W_V a_i$$

The key matrix $K \in \mathbb{R}^{l_a \times d_k}$ and value matrix $V \in \mathbb{R}^{l_a \times d_v}$ can be assembled by stacking the k_{a_i} and v_{a_i} vectors, respectively. To allow the attention mechanism to work, the dimension sizes d_q and d_k must match, as the dot product between query and key vectors requires aligned dimensions.

The actual (soft) attention is defined as the following function

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

where $QK^T \in \mathbb{R}^{l_b \times l_a}$ represents the unnormalized attention scores between the sequence of context vectors b_i and the input sequence a_i.

Which value vectors v_i from the input vectors should be primarily propagated further is determined by attending to the context sequence, and thus, weighting the value vectors v_i by the *similarity* between the key vectors and the query vectors from the context vector sequence. In the case of an attention mechanism using the *scaled dot product*, the dot product is used as the similarity measure, and $\sqrt{d_k}$ serves as a normalization factor.

There are two major kinds of attention: *cross-attention* and *self-attention*. In the case of cross-attention, the sequence of context vectors b_i and the sequence of input vectors a_i are different sequences. In the case of self-attention, they refer to the same sequence where $l_a = l_b$ and $a_i = b_i$ for $i = 1, 2, ..., l_a$.

2.2.2 The Transformer Architecture

The *transformer* architecture was introduced by Vaswani *et al.* [95] in the context of neural machine translation. The architecture heavily relies on cross- and self-attention

in order to generate an output sequence for a given input sequence. Although the initial architecture was designed for machine translation, modifications to this initial transformer architecture have been made to adapt it for other purposes as well.

Due to its initial design for machine translation, it is assumed that two sequences exist. The first sequence is fixed and fully observable as an input sequence to the model, such as the token embeddings of the sentence in the source language. This sequence of embeddings, a term for semantic dense vector representations, is processed by the *encoder* stack. The second sequence represents the sequence of embeddings of the already generated tokens of the sentence in the target language. However, because it has no information on upcoming tokens for obvious reasons, all such upcoming tokens are masked. The second sequence is processed repeatedly by the *decoder* stack.

While the transformer architecture in theory can handle input sequences of arbitrary lengths, its maximum input length is fixed and input sequences are padded to a homogeneous size in order to allow faster processing in fixed batches even if the input sequences differ in length, as well as due to practical considerations such as performance constraints.

In addition to the attention mechanism, the following concepts are fundamental to the transformer model.

Positional Encoding In contrast to RNNs that encode the notion of sequence ordering through their recurrent digestion of an incoming data sequence, the transformer architecture is a feed-forward model without an intrinsic token-wise positional awareness. Therefore, a positional encoding is usually injected at each token position to provide information about the position of a token in a sequence to the model. The original transformer model uses sinusoidal embeddings, however, other strategies like learnable positional embeddings also exist. However, since these positional encodings are typically designed for a fixed maximum sequence length, they prevent the model from effectively handling sequences longer than this predefined limit.

Multi-head Attention The attention mechanism as described in the previous section can be further stacked to increase the capacity of a single attention component structure, also referred to as *attention head*. Instead of passing the output of a single attention head to the next layer, multiple different attention heads are trained in parallel and their outputs are concatenated and further transformed into a reduced output vector by a learnable weight matrix that performs a linear operation. This constitutes the *Multi-head attention* concept and is shown in Figure 2.2.

Masked Attention As described earlier, because the model does not have access to all tokens until the end when all tokens have been predicted, the token positions of yet unknown tokens must be masked in order to deny the model from attending to such positions. The masking is implemented in the attention mechanism when the scaled dot-product is computed to obtain the similarity scores. For all yet unknown

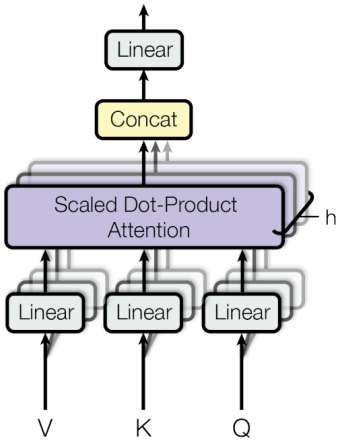

Figure 2.2: Illustration of a Multi-head attention block. (Graphic from Vaswani *et al.* [95])

token positions, the scaled similarity scores in the attention map are set to $-\infty$ before the Softmax operation and thus, this ensures that the final attention weights ignore yet unknown token positions. Figure 2.3 visualizes the masking.

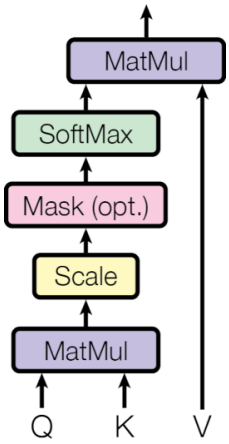

Figure 2.3: Optional masking of attention prior to the Softmax operation. The masking is required when unknown token positions should be suppressed. (Graphic from Vaswani *et al.* [95])

The whole transformer architecture is depicted in Figure 2.4 which is composed of the previously described elements. The token sequence of the first input sentence, e.g. a sentence in the source language in machine translation tasks, is regarded as the (fixed) inputs that are transformed into embeddings and fed into the *encoder* stack along with their positional encodings added to the embeddings. The *decoder* stack is fed with the token embeddings that were predicted during previous iterations, which also include the positional encodings for each token position from previous iterations. In machine translation, this corresponds to a sentence in the target language. The ultimate objective of the transformer model is to predict the next token for a known token input sequence.

Both concepts of encoder and decoder are fundamental and thus described further.

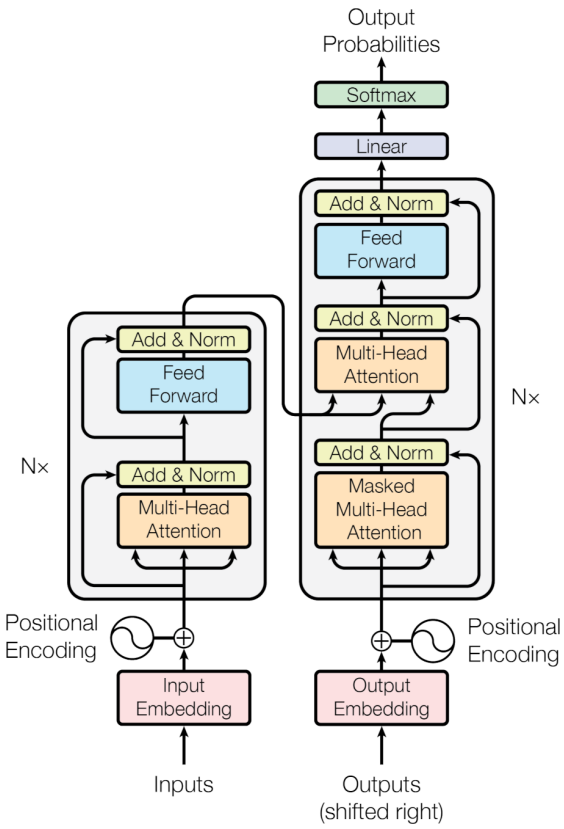

Figure 2.4: The transformer architecture. The bottom red boxes refer to the token embedding. In the center, the left box component describes the encoder part and the right box component describes the decoder part. The top two minor boxes illustrate the language modeling head. (Graphic from Vaswani *et al.* [95])

Encoder Block The encoder block consists of two main components. First, the multi-head attention component employs the self-attention paradigm that allows the attention heads to attend to any other intermediate token embedding in bidirectional ways. This means that at a certain token position the attention can include information from both left and right token positions. Second, the outputs are transformed by the second component, a dense feed-forward layer. In both cases, residual skip-connections as well as layer normalization are applied. One encoder block is usually stacked successively to generate the final contextualized token embeddings as the output of the entire encoder stack. The outputs are passed to all blocks of the decoder stack. Since the encoder input remains fixed during the subsequent token generation process, the output embeddings are only computed once by the encoder.

Decoder Block The decoder block mainly comprises the masked multi-head attention block, a multi-head attention block for cross-attention purposes, and a final dense feed-forward layer. Similar to the encoder block, a residual skip-connection and layer normalization are added in all three cases. Contrary to the design of the encoder, the masked attention prevents the attention mechanism from attending to positions to the right of a given position, and hence no bidirectional information flow is possible but for a certain position, only information from the left position can be attended to. However, the masked multi-head attention block uses a self-attention mechanism.

The outputs of the masked multi-head attention block are further processed by the multi-head attention block that employs cross-attention to the final embedding outputs from the encoder stack. Thus the information from the fixed inputs is injected into every decoder block of the decoder stack. As the last step, similar to the encoder block, the decoder block applies a final dense layer to predict a final output embedding of the decoder block. Multiple decoder blocks can be stacked in depth to one larger decoder stack. The final token representation is used to predict the probability distribution over the next token via the final dense layer with a Softmax activation function.

The actual, next token is determined by a specific token decoding strategy that samples the next token from the predicted probability distribution. For instance, in *greedy decoding* strategies, the next token is selected by the *argmax* operation on the probability distribution. If the next token does not yield a special stop token, also referred to as the *end of sequence* (EOS) token, and the maximum sequence length of the transformer model has not been reached, that token is again fed into the decoder stack to attend to previously generated tokens as well as to the initial input sequence through the encoder in order to predict the next token in an iterative fashion.

Both fundamental concepts of encoder and decoder blocks from the transformer architecture have been adapted for other tasks as well.

2.2.3 Encoder-only Masked Language Models

The transformer architecture is mainly designed for sequence-to-sequence tasks such as machine translation, and features a rather complex structure which may not be required in applications such as text classification or NER. Therefore, other, more simplified architectures have been proposed to better address these cases. One of the most notable architectures is the *BERT* architecture, an encoder-only model [96]. As indicated, the model rather relies on an encoder stack to perform certain NLP tasks, and the following points are characteristic of the concept:

- **Joint Bi-directional Encoding**: The encoder-only model applies self-attention at every encoder block. However, all tokens are available to the self-attention mechanism and at a certain token position it can attend to other, both left and right token positions, hence, it is referred to as bi-directional encoding. In addition, the computation of the attention vectors as well as all intermediate embeddings can be computed jointly in a simple, feed-forward style.
- **Masked Pre-training**: Due to its less complex structure, the pre-training of the network is simplified as it can be trained on raw sentences in a self-supervised way. The distinctive pre-training objective concerns *masked language modeling*. Hereby, a certain subset of the input tokens are not directly fed into the encoder but are masked such that the network cannot directly gain information on these token positions. This is implemented by feeding a special *[MASK]* token into the network instead of the actual token if that token is subject to masking. By training the model to reconstruct the masked tokens through a language modeling head on

top of the output embeddings only from its context, the model learns to generate contextualized token embeddings that can express semantically meaningful dense vector representations.

Several related approaches such as RoBERTa [97] have been developed that introduce various changes to the original BERT model, for instance, dropping the *next sentence prediction* task during pre-training. Due to the capabilities of the masked language model approach with an encoder-only architecture to yield contextualized, semantically expressive token embeddings, it is commonly used for text or token classification tasks. However, in classical text generation tasks, other approaches are typically preferable.

2.2.4 Decoder-only Autoregressive Language Models

While encoder-only models feature a bi-directional view of the token processing, decoder-only models pose an alternative approach to language modeling. As decoder-only models cannot attend to upcoming token positions but only to previous token positions, they are primarily designed to predict an embedding that is used to generate the next token distribution and are less common for tasks such as text or token classification. Moreover, the unidirectional approach expresses the auto-regressiveness of the problem statement of text generation, in which the distribution over the next token depends on the sequence of previous tokens. Because of that, decoder-only language models are commonly described as *autoregressive language models* or *causal language models*.

The various iterations of OpenAI's Generative Pre-Trained (GPT) transformer models [98, 99, 73] are popular examples of such decoder-only models and therefore the term *GPT* is often used as a synonym for such autoregressive language models. Other similar open-source models follow the autoregressive paradigm as well, including notable LLMs [100, 101, 86, 87].

2.3 Neural Named Entity Recognition Modeling

When addressing the NER task using neural methods, certain structural concepts are commonly shared across various implementations, although nuanced differences in implementation details may exist. In this section, the generic overall architecture for NER, as applied in this work, along with two NER-specific sequence tagging schemes, are introduced.

2.3.1 Architecture for Token Classification

The NER task is commonly framed as a token classification task and can therefore be addressed by an ordinary model architecture featuring two key components: the encoder and the NER head. A new input text is initially tokenized into a set of tokens. However, this step depends on an encoder-specific tokenizer and vocabulary to obtain the vocabulary-specific token identifiers. The tokens are subsequently processed by the

encoder to generate token-wise contextualized dense vector representations, referred to as token embeddings. The encoder can be implemented simply as a feed-forward network from scratch or as a more complex transformer network initialized with pre-trained weights. The second key component, the NER head, processes the token embeddings to predict classification scores at each token position. Similar to the encoder, the actual implementation details of an NER head can vary. Both the NER head as well as the encoder are trained through backpropagation. While the encoder may be used with pre-trained, and potentially frozen weights, the NER head usually requires training from scratch. The overall architecture is shown in Figure 2.5.

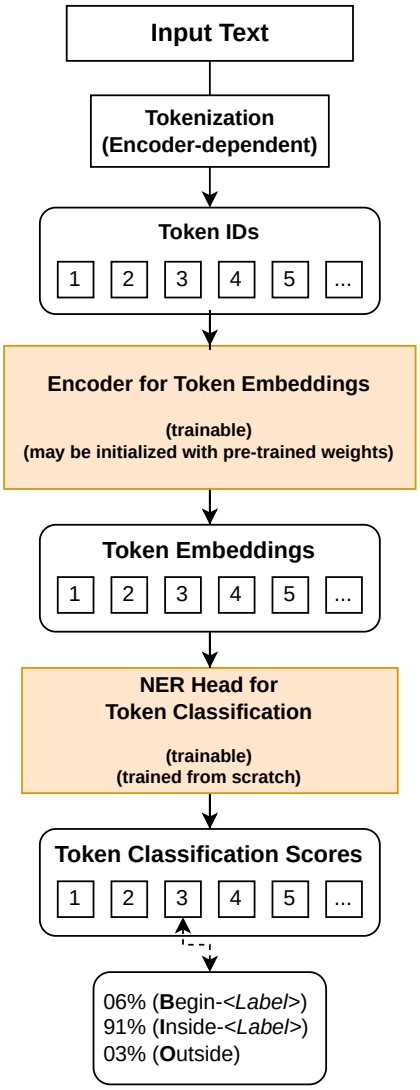

Figure 2.5: A generic token classification architecture commonly used for NER. The components in orange, namely the encoder and NER classification head are trainable via backpropagation.

2.3.2 NER Tagging Schemes in Sequence Labeling

In contrast to basic token classification tasks, in the context of NER, individual entities within a text are defined as label class-wise spans over one or more tokens rather than assignments of plain label classes to individual, independent tokens. To encode

entities, given in their span-oriented structure, into a sequential representation and subsequently reconstruct the encoded entities from a given sequence, a certain tagging scheme is employed. This allows the token classification framework to be re-purposed for predicting span-oriented entities.

For a set of n label classes, a tagging scheme further expands each initial label class into a set of action-prefixed label classes serving as sequence tags. These tags express certain sequence states in addition to the initial label class information. This idea is not unique to NER in particular, and has been also used for instance in generic text chunking contexts [102, 103]. In this context, the term *label scheme* may be used instead of the term *tagging scheme* in related literature.

Within the scope of this work, the following different tagging schemes are used.

- **IOB2** [102]: In IOB2, a label class is divided into **B-**<**Label**> (Begin) and **I-**<**Label**> (Inside) tags, accompanied by a label class-agnostic **O** (Outside) tag. Each token unaffected by any span is tagged as `O`. For a span with a certain label class, all tokens within the span are tagged as `I-<Label>`, except for the first token of the span which is tagged as `B-<Label>`. Occasionally, IOB2 is also referred to as BIO.

- **BILOU** [104]: BILOU defines an extension of the IOB2 scheme. The rules remain identical, except for spans over single tokens, which are tagged as `U-<Label>` (Unit), and for spans over multiple tokens, in which the last token is tagged as `L-<Label>` (Last). Occasionally, BILOU is also referred to as BIOES (**L**ast $\rightarrow$ **E**nd, **U**nit $\rightarrow$ **S**ingle).

- Apart from the previously mentioned, span-oriented entity encoding schemes, that can accurately define the begin and end of a certain entity, an annotation sequences may also dismiss this explicit notion, and stick to the raw label class reference without indication about the begin or end of an entity. This kind of tagging scheme is sometimes referred to as **IO**, albeit IO may also imply that an active token may be prefixed by `I-<Label>` (Inside). However, it conceptually matches a simple, non-prefixed tagging implementation. In this scheme, neighboring entities of the same label class cannot be reconstructed from an annotation sequence since information about the begin and end of entities are partially lost.

As a consequence of the tagging scheme encoding, its resulting sequence can only represent entities without overlapping spans.

Apart from the IO tagging scheme, only the two popular tagging schemes, as described earlier, are discussed in this work. However, other and more obscure tagging schemes exist as well. Nevertheless, some instances are almost identical to other tagging schemes with only subtle differences applied [103, 105, 106].

Given a certain tagging scheme, a set of entities can be transformed into a tagged sequence, as illustrated in Figure 2.6 for IOB2 and BILOU.

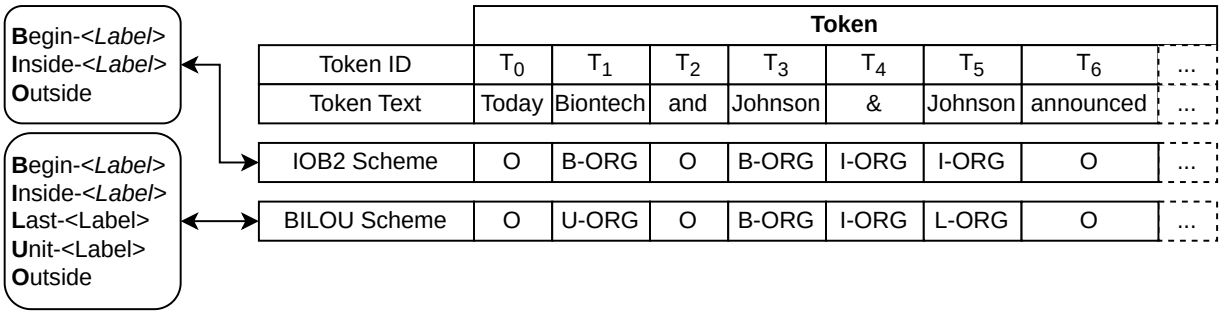

Figure 2.6: Illustration of the IOB2 and BILOU schemes for NER sequence labeling for the two *ORG* entities [*Biontech*] and [*Johnson & Johnson*].

2.3.3 Structured Prediction for Sequence Labeling

In order to decode the token classification scores, that are predicted by the NER head at a certain token position, into a valid tagging sequence, the actual tags are usually obtained using *structured prediction*. In NER, this technique aims to mitigate the sampling of sequences that do not follow the formal structure of a certain NER tagging scheme. For instance, in the case of an NER task with one label class and an IOB2 tagging scheme, the NER tag `I-<Label>` is considered invalid if the sampled NER label for the previous token is `O` because the tag of the first token of a label span must start with the `B-<Label>` according to the IOB2 scheme.

To address this challenge, relying solely on the raw token classification training with strictly valid tagging sequences only mitigates the model's behavior to predict valid label sequences but does not eliminate such risk. To ensure that only valid NER tags are obtained at a certain token position, all formally valid NER tags are determined based on the sampled NER tags at the previous token position. Using all valid NER tags, all invalid NER tags from predicted token classification scores are masked prior to the final NER tag sampling. The final NER tag sampling can be implemented with different strategies such as greedy decoding or beam search. Within this work, greedy decoding is used due to its simplicity and computational efficiency. Greedy decoding directly selects the NER tag from the token classification scores using argmax. The annotation sequence, consisting of the individual entity spans along with their corresponding label classes, can be safely reconstructed because the sampled tagging sequence is ensured to be valid with respect to a particular tagging scheme.

The process of the structured label decoding is shown in Figure 2.7. Terminology-wise, structured prediction can be also referred to as *structured generation*, *structured decoding* or *constrained decoding* in related contexts. The terms are used interchangeably in this work, yet structured prediction tends to be more generic.

2.3.4 Evaluation Approaches of Tagging Sequences in Named Entity Recognition

Measuring the correctness of a certain annotation sequence with respect to its ground truth counterpart can be an ambiguous task, and thus, multiple strategies for com-

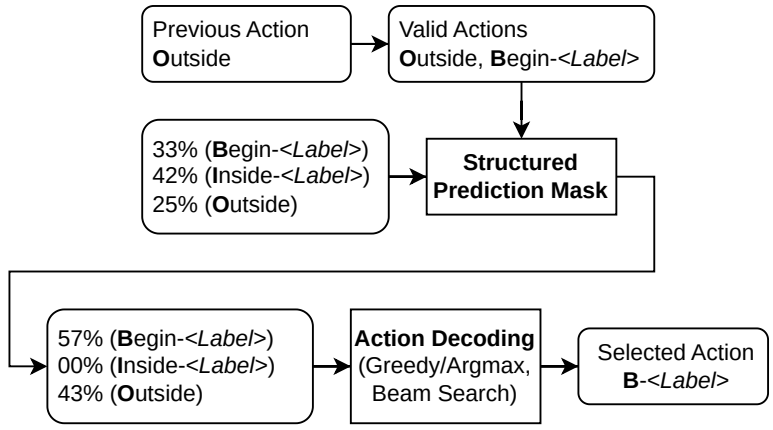

Figure 2.7: Illustration for structured prediction in NER sequence labeling tasks during sequence decoding.

paring annotation sequences can be employed. The choice of evaluation strategy can critically influence the final evaluation scores, and also has consequences in terms of comparability. As measures like precision, recall and F1-score are used in NER, defining incorrect and correct annotations of entities is in particular crucial for computing these measures. Two fundamental differences are worth highlighting.

Token-wise evaluation The annotation data can be evaluated by comparing the assigned token tags with the token tags from the ground truth at each token position independently. The counting of incorrect or correct tags can be subsequently used to determine precision, recall, and F1-score. Since the annotation sequence is encoded according to a certain tagging scheme, different tagging schemes may lead to different final scores for identical entities if divergences between the ground truth and the prediction are observed. These effects are illustrated in Figure 2.8.

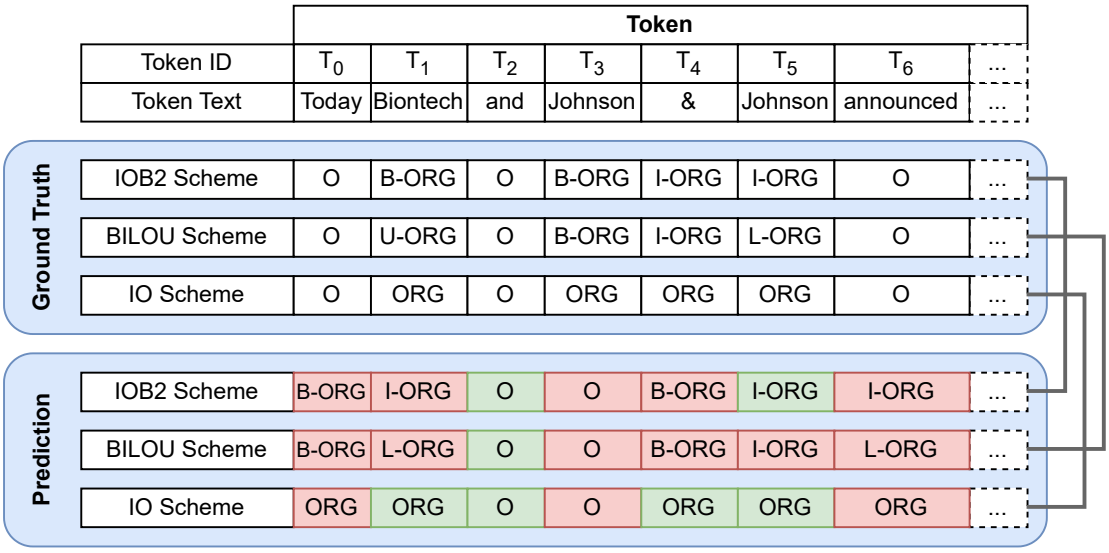

Figure 2.8: Illustration for the token-wise evaluation strategies with different tagging schemes. The underlying, predicted entities slightly differ from the ground truth. ([*Biontech*] → [*Today Biontech*] (expansion left), [*Johnson & Johnson*] → [*& Johnson announced*] (shift to right)) The green cells indicate correctly predicted token tags and the red cells indicate incorrectly predicted token tags, according to their corresponding ground truth cells.

While sequence tagging uses a token-aligned structure, comparing a certain method to differently pre-tokenized ground truth data may introduce inaccuracies if a re-tokenization into a common token sequence structure is required and spans might not align well onto the updated token boundaries. To mitigate this issue, a character-wise evaluation of the sequence tags can be used instead.

Entity-wise evaluation In contrast to counting the matches and non-matches in pairs of prediction and ground truth token tags at each token position individually, an entity-centric counting strategy poses an alternative for determining precision, recall and F1-score.

Given the set of predicted entities and ground truth entities, each (prediction, ground truth)-pair is verified for a *match*. Because this approach may include situations that encompass overlapping entities within the prediction or ground truth set, the handling of overlapping entities is implementation-dependent. For the case of the evaluation script from the n2c2 2018 ADE challenge [107], if multiple predicted entities can be matched by the same ground truth entity, the ground truth entity is counted as correctly detected once, and the other predicted entity spans that match that ground truth entity are not counted as false positives. This also avoids situations in which multiple predicted, non-overlapping entities are be matched with single ground truth entities and lead to ambiguous mappings between potential entity correspondences. For the sake of simplicity, cases in which multiple predicted entities match a single ground truth entity are ignored in this explanation.

The number of true positives entities (#TPs) is given by the number of ground truth entities that are matched by at least one predicted entity. The number of false positives (#FPs), or false negatives (#FNs), can be determined by the number of predicted, or ground truth entities, subtracted by #TPs, respectively. Due to the design of this approach, no true negative counts exist (#TNs = 0). The precision, recall and F1-score measures can be directly inferred from the resulting confusion matrix.

Depending on the strictness of the evaluation, a match between a predicted entity and a ground truth entity with an identical label class is defined differently. Common definitions include:

- **Strict Match**: For strict matching criteria, the spans of the entities need to match exactly. This mode is also referred to as *exact* span matching.
- **Lenient Match**: The lenient matching criteria allows entities to match even if their spans are not identical. It is also referred to as *inexact* span matching. This means that entities are considered to match if their spans just share some common overlap. While this behavior allows for *any* minimal overlap to suffice as a valid match, some implementations also offer an additional constraint to only match entities if a certain span-related similarity threshold is passed.

The entity-wise evaluation strategy is used in several implementations, such as the

n2c2 2018 [107] and 2022 [108] evaluation scripts[1], the NER scorer of Spacy [85] (version 3.7.5)[2] or the CoNLL 2000 [109] evaluation script[3]. The CoNLL 2000 perl implementation has been re-implemented by the Python library *seqeval*[4] [110] which is commonly used by Huggingface's Transformers library[5] [111].

In general, among the various evaluation techniques for obtaining precision, recall, and F1-score, no single technique can be considered universally superior to the others. Depending on the nature of the target entities, certain assumptions may be relaxed or must remain strict. However, no reliable conclusions can be drawn from comparing scores if these scores are calculated using different evaluation strategies.

[1] https://github.com/Lybarger/brat_scoring/blob/cab26377a7c373eacbb36073885183ab468158a1/brat_scoring/scoring.py (accessed December 1st, 2024)

[2] https://github.com/explosion/spaCy/blob/a6d0fc3602b611e4d4a6fcc4f41cbea114ad13f8/spacy/scorer.py#L760 (accessed December 1st, 2024)

[3] http://www.cnts.ua.ac.be/conll2000/chunking/conlleval.txt (accessed December 1st, 2024)

[4] https://github.com/chakki-works/seqeval (accessed December 1st, 2024)

[5] https://github.com/huggingface/transformers/blob/5d7739f15a6e50de416977fe2cc9cb516d67edda/examples/pytorch/token-classification/run_ner.py#L529 (accessed December 7th, 2024)

Chapter 3

Materials

3.1 Text Corpora

In general, having access to text corpora is essential in the process of building NLP models, as their text samples can be directly used as training resources, but also indirectly for validation or development set as data-backed material to guide further decision-making during the development and training phase. Even after the development of an NLP model, text corpora are important for an independent evaluation. However, as mentioned in previous chapters, the difficult situation in low-resource languages and domains remains a major obstacle [48, 49].

This section concerns text resources and corpora that are closely related to health information or the (bio-)medical and clinical domain. While not all corpora that are mentioned in this section are directly utilized by this work, they are highly relevant in the broader context of this work. On a global level, the number of available corpora in various languages exceeds the scope of this work, and the interested reader is referred to a recent, explicit survey on datasets in different languages [49]. Instead, only a selected subset of relevant corpora are discussed in the following subsections. For a comprehensive overview of the German medical and clinical corpora, the reader may refer to Hahn [112].

3.1.1 English Corpora

MIMIC-III MIMIC-III [36] is not primarily a text corpus, but a database consisting of several data sources from the intensive care unit (ICU) of the Beth Israel Deaconess Medical Center in Boston, Massachusetts. The database includes several kinds of medical signal and monitoring data, lab and microbiology test data, billing and accounting information as well as clinical notes of discharge summaries. The authors applied de-identification, date shifting and other data transformations to the dataset in order to grant other researchers access to the data.

In the context of text corpora, MIMIC-III contains 59,652 discharge summaries in English. Due to the de-identification process, privacy-related text entities are masked to mitigate re-identification risks, but otherwise, the actual text samples remain authentic.

However, the text notes are largely unannotated and hence, lack information on potential entity classes or relation information.

Other iterations of MIMIC, namely the predecessor MIMIC-II, and its successive work MIMIC-IV [37] exist as well. While MIMIC-II is replaced by MIMIC-III and MIMIC-IV according to its authors[1], MIMIC-IV is not yet as popular as MIMIC-III due to its rather recent publication date.

n2c2 2018 ADE Track 2 In 2018, the *National NLP Clinical Challenges* (n2c2) hosted two scientific challenge tracks [107]. The second track involves the detection of adverse drug events (ADE) and medication extraction in clinical documents. Along with the challenge, its authors have created and are providing 505 discharge summaries from the MIMIC-III dataset with their corresponding annotation information on several label classes as well as relation information between certain entities. While the initial focus of the track lies on ADEs, other entities with no direct affiliation to ADEs are annotated as well. Due to its MIMIC-III origin, the discharge summaries are in English. The label classes consist of `Drug`, `Strength`, `Reason`, `Route`, `Form`, `Duration`, `Dosage`, `Frequency` and `ADE`. Furthermore, relation information between `Drug` and the other label classes is provided.

3.1.2 German Corpora

Bronco150 The *BRONCO150* dataset [34] consists of text from 150 discharge summaries from the oncology departments at Charité Universitätsmedizin Berlin and Universitätsklinikum Tübingen. All of these 150 discharge summaries were anonymized manually. The actual text corpus comprises five virtual documents that are composed of a set of sentences that have been selectively extracted from these 150 discharge letters in order to counter potential de-anonymization attempts. Therefore, the documents do not carry context information beyond the scope of each single sentence. The text samples are authentic instances from the clinical oncology domain. The label classes are `Medication`, `Treatment` and `Diagnosis`, but also ATC [113], ICD-10 [114], and OPS [115] code groundings for certain entities are part of the annotation data.

GGPOnc The GGPOnc corpus [64, 32] is composed of a set of documents of clinical guidelines for oncology drawn from the *Leitlinienprogramm Onkologie* published by the *Arbeitsgemeinschaft der Wissenschaftlichen Medizinischen Fachgesellschaften e.V. (AWMF)*, *Deutsche Krebsgesellschaft e.V. (DKG)* and *Deutsche Krebshilfe*. Because these documents do not contain any information or references to individual patients nor do they conflict with any privacy-motivated policies, the obtained corpus bypasses common legal obstacles and hence, it is publicly available. Two iterations of the corpus have been published, the initial version as well as a follow-up version that includes a larger corpus (+40%) with updated annotation guidelines and label class schemes that conform well with the SNOMED CT [116] top-level hierarchies. The

[1] https://mimic.mit.edu/docs/ii/ (accessed February 22th, 2024)

first iteration covers 25 topics from the German oncology clinical guidelines, but newer versions on the same topics as well as five new topics were added to the second iteration. For the first iteration [64], a pre-annotation was created with the UMLS metathesaurus [117] automatically. For a subset of the corpus, the annotations were manually updated and subsequently framed as gold standard quality. The entity classes from the first iteration are `Living_Beings`, `Procedures`, `Disorders`, `Physiology`, `Anatomical_Structure`, `Chemicals_Drugs`, `TNM` (referring to TNM Classification of Malignant Tumors), and `Devices`. The data is provided in the common *BRAT* file format. The second iteration [32] composes a more diverse set of annotation schemes, with two determining parameter sets regarding span sizes (*short* and *long*) and label class sets (*coarse* and *fine*). For instance, a long span *zahnärztliche und ärztliche Untersuchung* is reduced to *Untersuchung* in the short span configuration. For the second parameter, the label classes can be chosen in a fine as well as in a coarse setting, leading to the label classes `Finding`, `Substance` and `Procedure` in coarse mode, and to the label classes `Clinical_Drug`, `Diagnostic`, `Nutrient_or_Body_Substance`, `Therapeutic`, `External_Substance`, `Diagnosis_or_Pathology`, and `Other_Finding` in the fine mode. The annotation data is provided in the pre-tokenized *CoNLL* text formats and *WebAnno*, which is a native file format of the *WebAnnotate* [118] tool. Moreover, the annotation data is also provided in binary Spacy and Huggingface formats. Furthermore, the addition of new annotation layers that may include SNOMED CT groundings is planned according to the authors.

CARDIO:DE The CARDIO:DE corpus [33] features 500 German doctor's letters from the cardiovascular domain and originates from the Universitätsklinikum Heidelberg and are therefore authentic texts. Similar to previous corpora, the documents have been anonymized with regard to the PHI information, but the document structure has been preserved. The corpus is accompanied by annotation information on several medical entities as well as the tagging of entire text sections. The covered entity classes are `ActiveIng`, `Drug`, `Duration`, `Form`, `Frequency` and `Strength`. The text sections concern `Abschluss`, `Anamnese`, `Anrede`, `Diagnosen`, `AufnahmeMedikation`, `Befunde`, `EchoBefunde`, `AktuellDiagnosen`, `EntlassMedikation`, `KuBefunde`, `Labor`, `Anderes`, `RisikofaktorenAllergien`, and `Zusammenfassung`. Similar to GGPOnc, the dataset annotations are provided in the WebAnno format from the WebAnnotate tool which is the predecessor to the latest *INCEpTION* [119] annotation tool that has been used for the CARDIO:DE study. The dataset encompasses the subcorpora *CARDIODE400* and *CARDIODE100_heldout*. The former subcorpus contains 400 documents along with its annotation information on medication and section spans. The latter subcorpus contains 100 documents but the annotation information is held back by the authors' intention to organize events on shared tasks and medical information extraction challenges. Along with the core CARDIO:DE corpus, the authors also provide an alternative, experimental annotation set that covers more entity classes, namely `Dosage`, `Reason` and `Route`. The authors justify the decision to keep these three classes apart from the core annotation data by the low inter-annotator agreements.

3.1.3 Multilingual Corpora

Mantra GSC The Mantra GSC (Gold standard corpora) corpus [65] originated from the *MANTRA*[2] project and consists of several sub-corpora in multiple languages. Hereby, the multilingual texts are mostly translations from the English source, and hence, parallel corpora. Each text sample consists mostly of a single, independent sentence or text snippets rather than covering larger documents. Three sub-corpora are available in English, Spanish, German, French, and Dutch:

- **Medline**: The Medline corpus is created of abstract titles from Medline. In total, 100 samples are part of this corpus.
- **EMEA**: This corpus encompasses drug labels that stem from the European Medicines Agency (EMA, sometimes called EMEA inofficially). In total, 100 sentence samples are included.
- **Patent**: 50 samples are patent samples that were drawn from the IFI Claims[3] platform. Each sample spans a text paragraph of multiple sentences. In contrast to both former sub-corpora, no Dutch data is provided.

All three sub-corpora are annotated by the entities `ANAT` (Anatomy), `LIVB` (Living Beings), `OBJC` (Objects), `DISO` (Disorders), `PROC` (Procedures), `CHEM` (Chemicals and Drugs), `PHEN` (Phenomena), `PHYS` (Physiology), `DEVI` (Devices), and `GEOG` (Geographic Areas). The annotation information also includes grounding information to UMLS entries via a CUI reference.

3.2 Knowledge Bases and Related Resources

3.2.1 Wikipedia and WikiText

The open encyclopedia Wikipedia is commonly used as a key source for obtaining high-quality text resources in a cheap yet reliable fashion. For instance, it is used for pre-training various BERT models, e.g. the German *bert-base-german-cased* monolingual BERT model[4]. In addition to the online HTML contents from its web platform, the Wikimedia foundation publicly provides date-ordered snapshots of all relevant data in a compressed and machine-readable format[5].

Within the context of this work, the Wikipedia resource structure is simplified according to the following definitions. A Wikipedia resource instance is a *language-dependent* set of *pages*. While references across different languages exist, Wikipedia instances of two different languages are still conceptually isolated instances. Each page can be considered as a document along with an associated page title. Each document itself consists of the text of the page encoded in the WikiText markup language to

[2] https://cordis.europa.eu/project/id/296410 (accessed March 19th, 2024)
[3] https://www.ificlaims.com/ (accessed March 20th, 2024)
[4] https://www.deepset.ai/german-bert (accessed April 8th, 2024)
[5] https://dumps.wikimedia.org/ (accessed April 8th, 2024)

include meta information such as references ("links") to other Wikipedia pages, section headings or info boxes, in addition to the raw text itself.

Furthermore, contrary to usual page instances, certain pages are flagged as redirections to other pages and hence, lack individual content except for the redirection information on the target page. Besides the main set of Wikipedia pages, consisting of either WikiText articles or redirections, Wikipedia has pages from other *namespaces*, that host non-article pages such as user discussions, drafts or template pages. In this work, only pages from the common article namespace are used in the text processing of the Wikipedia corpus.

WikiText is part of MediaWiki, a PHP-based web platform for hosting Wiki-like web-pages such as Wikipedia, and therefore provides a reference implementation to convert WikiText into an HTML representation. Therefore, the reference implementation is inept to obtain the plain text data along with the metadata information. Alternative parser implementations may be preferred as they often support the conversion into a plain text format well suited for furher text processing purposes.

In this regard, there only exist one alternative parser that features a complete WikiText implementation, and several other parsers with incomplete support[6]. The mentioned, complete implementation *Parsoid*[7] is build for HTML-related use cases and can establish a bijective mapping between a valid WikiText representation and its corresponding HTML representation. It's intended purpose is to be used by MediaWiki to provide a modern, user-friendly, web-based WikiText editor. Other parsers, albeit with incomplete implementations, therefore may still be preferred depending on the particular purpose, in particular with respect to their ability to extract plain text from the WikiText data.

3.2.2 WikiData Knowledge Base

Representing knowledge in a structured, machine-readable and machine-interpretable format may depend on the actual knowledge semantics as well as the underlying motivating objective, yet it is crucial for various applications in artificial intelligence such as expert systems and machine reasoning. Similar to the Wikipedia platform that provides information encoded as unstructured natural text, a popular and active counterpart in the structured knowledge representation realm is the WikiData knowledge base [120]. It is an open platform for creating and maintaining structured data on global knowledge. Contrary to language-dependent Wikipedia instances, the WikiData knowledge base itself is designed as a language-agnostic knowledge base, but language-specific terms and references can be part of certain data entries.

For the sake of the scope of this work, only a simplified view on the actual structure is presented. Here, the WikiData knowledge base can also be in particular considered as a knowledge graph due to its Resource Description Framework (RDF)-oriented,

[6] https://www.mediawiki.org/wiki/Alternative_parsers (accessed April 8th, 2024)
[7] https://www.mediawiki.org/wiki/Parsoid (accessed April 8th, 2024)

graph-like structure. The graph is mainly composed of node-like entity items as well as typed and directed edges between two nodes.

Each entity is identified by a unique id ("QID", e.g. Q18216 for entity *aspirin*) and a set of language-specific label texts, that are used as referring term in a certain target language. To take into account that alternative terms like synonyms or informal language may be common for certain entities, language-specific sets of alias terms may be assigned. In addition, short language-specific description texts can be assigned to an entity to further add basic information and disambiguation hints in addition to the entity label.

The actual structured information on an entity is composed of a set of *statements*, also referred to as *claims*. As every entity is also a node in the knowledge graph, every statement is defined via a relation between an entity and a data target, encoded by a typed, directed edge, referred to as *property*. Similar to entity nodes, each property type is uniquely identified by a number with the "P" prefix (e.g. P31 refers to the *instance of* property). The data target can either be an actual entity, represented as another node in the WikiData graph structure, or pure factual information that comprises data such as dates, numbers or coordinates.

Moreover this format structure is also applied in order to encode external identification descriptors, referred to as *identifiers*, about an entity. Given a certain external coding system and its corresponding code for the entity, that code is assigned to the entity as a factual information through a property that is dedicated to this external coding system.

An entity may be augmented by a set of sitelinks to other related pages, in particular language-dependent Wikipedia pages and other Wikimedia-affiliated pages such as Wikiquote. Therefore, a sitelink describes the reference of an entity to its semantically corresponding instantiation beyond its WikiData representation.

The actual entities in WikiData not only cover typical subjects similar to Wikipedia page subjects, but also encompass abstract concept entities that shape the graph into hierarchies and ontology-like structures beyond a generic knowledge graph structure.

3.3 Software and Tooling

3.3.1 Full-text Search using Apache Solr

Fast full-text search is a fundamental task in common applications that need to deal with longer textual passages or a large number of short texts. Here, the problem becomes challenging in such cases, as simple brute-force approaches that scan the entire text without optimization measures can hit performance limits in terms of latency and responsiveness. Relational databases support fast access to database cells in sophisticated ways for schema-aligned data via indices. However, support for plain text search depends on how individual database engines are implemented, and different types of database engines may not necessarily support more complex search queries, such as *Levenshtein*-based similarity search for fuzzy search applications.

A prominent example of a full-text search engine implementation is Apache Lucene[8], a Java-based library that features several fundamental text search algorithms and basic acceleration techniques for faster query times. Lucene's document-oriented structure enables its use in building search engine implementations. Additionally, Lucene provides functionalities like tf-idf scoring and k-nearest neighbor search. It also includes text preprocessing operations such as basic tokenization, stemming algorithms, and stop word filtering that can be applied in combination with exact search.

Lucene combines this variety of functionalities into a software library, with API bindings to Python (PyLucene)[9], and serves as a core component for higher-level search engine platforms such as ElasticSearch[10] and Apache Solr[11]. Although ElasticSearch and Apache Solr follow different design choices, their differences are considered negligible in the context of this work.

Apache Solr is a search application that can manage multiple instances of indexed search tables. It builds on top of Lucene and conceptually resides on a higher abstraction level. Solr maintains multiple Lucene index tables, referred to as *collections*, which can be subdivided into smaller *shards* for distributed search. Moreover, Solr is configured by *configsets*, which define data indexing and preprocessing operations for one or multiple collections as well as deployment-specific settings. This also includes the definition of exposed REST endpoints and subsequent request handlers.

For the understanding of this work, the *Tagger Request Handler* serves an important role. The component can process an input text and return the set of terms in the text that match one or multiple indexed terms from a Solr collection, along with their locations in the text. In practice, this implies that a certain text entry can be tagged with several possible indexed term matches. While the tagging can be considered as a dictionary-based entity detection mechanism, it only provides a set of possible match candidates, and hence, a complementary entity disambiguation step in the follow-up postprocessing pipeline is required to enable an overarching NER or EL pipeline. Moreover, since the tagging mechanism resides in the realm of search-based approaches, it does not leverage sophisticated NLP methods which may lead to inappropriate matches with respect to false positive entity detections (e.g. "Leber" (anatomy, indexed) in "Herr Leber" (text snippet)) as well as false negatives (e.g. "Krebs" (disease, indexed) in "Krebs-Forschung" (text snippet).

3.3.2 Entity Linking using OpenTapioca

As discussed in the previous chapter, the EL task in NLP reaches beyond the detection of certain entities inside a text and aims to establish a reference mapping between a detected entity and its corresponding concept from a particular knowledge base. OpenTapioca [121] is an open, multilingual EL system that is designed

[8] https://lucene.apache.org/ (accessed April 15th, 2024)
[9] https://lucene.apache.org/pylucene/ (accessed April 2nd, 2024)
[10] https://www.elastic.co/ (accessed April 15th, 2024)
[11] https://solr.apache.org/ (accessed April 15th, 2024)

to solely work on WikiData as knowledge base target. Therefore, the entirety of the system is based on open-source and non-proprietary components. In contrast, the use of SNOMED CT knowledge base as an ontology for medical applications is well established and plays an elementary use in related medical standard protocols like HL7 FHIR, however, SNOMED CT remains a proprietary and non-community-driven product which imposes certain legal restrictions. Therefore, WikiData does not share similar copyright-related limitations.

From an architectural point of view of an EL system, the processing logic is usually comparted into three substeps in order to process an incoming text and enrich it with the semantic linking metadata:

- **Span Detection**: Given the text, all (potential) instances of terms in the text, that could pose a possible conceptual reference to a corresponding entity in a knowledge base, e.g. WikiData in OpenTapioca, are detected during that step. This step is closely related to NER, except that the entities do not necessarily require an explicit label assignment depending on individual EL implementations. In OpenTapioca, only the bare span information which defines the start and end of the entity remains relevant.
- **Candidate Generation**: Based on the previous detection of potentially relevant text spans, the peripheral knowledge base of the EL system is scanned for potential matches given the search term. The search term may be either kept identical to the text within the detected text span or modified in a certain way to optimize for an increase of the likelihood of finding the correct corresponding entity in the ontology, or even go beyond the static text search by employing alternative techniques such as vector-based nearest neighbor search on semantic embeddings. Each candidate describes the potential link between a certain text span and an entry from the knowledge base.
- **Candidate Resolution**: In this step, all possible candidates are scored according to a certain metric in order to determine the best candidate to match for each detected text span. This may also include the decision to discard all candidates of a certain text span and hence, reject the linking of such text span itself. Terminology-wise, candidate resolution is also called *candidate selection*, *re-ranking*, or *entity disambiguation*, although the term *entity disambiguation* may also include the candidate generation step as well.

In OpenTapioca, the first two steps, span detection and candidate generation, are implemented by the use of an Apache Solr setup along with its tagger request handler. For this, the Solr collection for OpenTapioca needs to be initialized first to contain all indexed terms. Given a user-defined set of WikiData entities, this is realized by gathering all labels and alias terms from all languages for each entity and indexing these terms along with their QID identifiers. Furthermore, certain identifiers can also be included into the index which can increase the effectiveness for the entity detection depending on the actual objective. For instance, the inclusion of user names from

social media can be beneficial for detecting mentions of public figures, or similarly, the inclusion of ICD [122] codes may help to detect mentioned diseases that are only referred to by their code in the text.

Due to the fact that Apache Solr is employed to address the first two steps of EL, the essential part of the OpenTapioca system concerns its algorithm design to approach the candidate resolution step. In this step, for each pair of detected text span and its corresponding entity candidate, a hand-crafted feature vector is assembled that takes the following features into account.

- **Prior probability of the text span**: To estimate the prior probability of the text within the detected span to appear in a document, this text span is compared to entries in a unigram language model that has been trained on the WikiData title and labels. Note that this feature item does not depend on the associated entity candidate.

- **Prior probability of the entity candidate**: Similar to the estimation of the popularity of the text span, also the popularity of the entity candidate is taken into account. Due to the fact there is no access to a large corpus with linked entity annotations, the page rank of the entity within the WikiData graph is used as an approximative value.

- **Number of statements**: By counting how many statements are associated with a certain entity, it can be an factor indicating the popularity of a possible entity candidate. Unlike page rank, it solely counts the number of outgoing edges from the entity to other nodes or data targets in the graph.

- **Number of sitelinks**: Likewise to the previous feature, also the number of sitelinks to other pages of Wikimedia platforms, such as to language-specific Wikipedia or Wikiquote pages, is included as a feature item.

- **Constant value**: A fixed value that remains constant for every feature vector.

Within the framwork of OpenTapioca, these features are used to estimate the *local compatibility* between a potential text span and its entity candidates. In practical terms, an example which features a set of detected text spans and their corresponding entity candidates is shown in Figure 3.1 for illustration purposes.

Instead of directly trying to use these local compatibility features in order to score and classify potential linkings between text spans and entities, OpenTapioca attempts to include context information and a certain degree of awareness towards entity coherence into the decision process. For OpenTapioca, this requires a metric to measure the semantic similarity between two entities from WikiData. According to the manuscript of Delpeuch [121] on OpenTapioca, the similarity metric, capable to operate on the

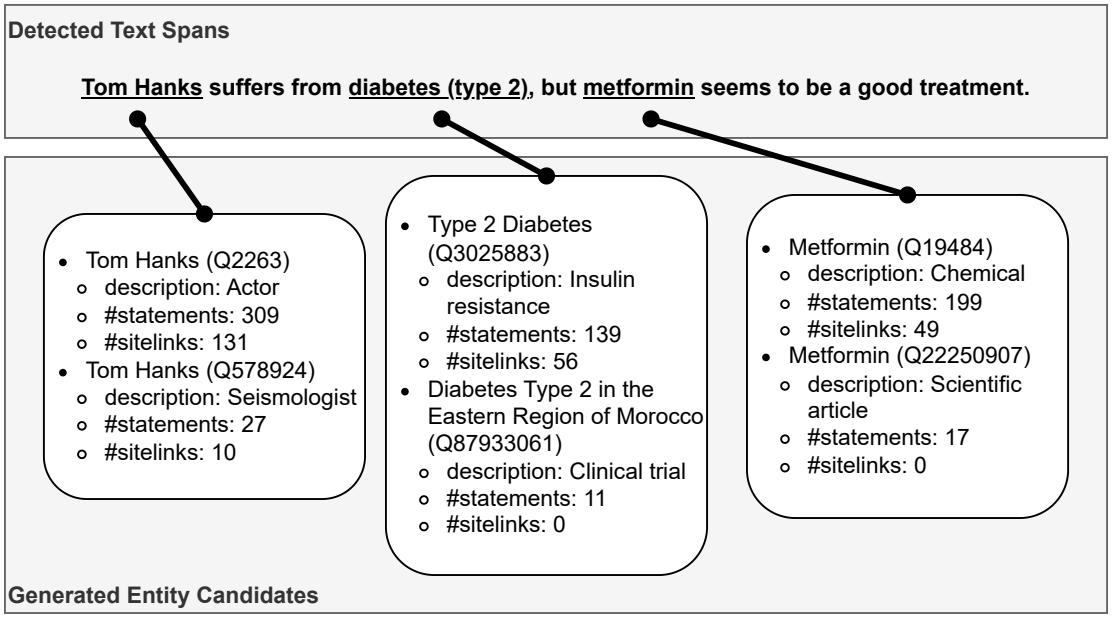

Figure 3.1: Example for span detection and candidate generation for a short sentence in OpenTapioca. An exemplary set of plausible, detected text spans (underlined) with their corresponding entity candidates are shown for demonstration purposes. The results serve only illustratory purposes and differ in reality.

graph structure, is defined as

$$
\begin{aligned}
sim(entity_1, entity_2) = & \beta^2 \delta_{entity_1 = entity_2} + && \text{(I)} \\
& \beta(1-\beta)\left(\frac{\delta_{entity_1 \in l(entity_2)}}{|l(entity_2)|} + \frac{\delta_{entity_2 \in l(entity_1)}}{|l(entity_1)|}\right) + && \text{(II)} \\
& (1-\beta)^2 \frac{|l(entity_1) \cap l(entity_2)|}{|l(entity_1)||l(entity_2)|} && \text{(III)}
\end{aligned}
$$

where $l(entity)$ yields the set of statements that point to other entities in the graph for a given entity, $\delta_{condition}$ refers to the Kronecker delta function that yields 1 if the given condition is met and 0 otherwise, and $\beta \in [0, 1]$ as a scalar hyperparameter. The semantic similarity is thus derived from the notion of *reachability* on the graph structure. It estimates the probability to end up at the same entity after at most one random edge transition from each initial nodes of $entity_1$ and $entity_2$, and given the probability β to omit a transition. This can be split into three cases: No transition happens at all (I), only one traversal at one of the two initial nodes along a statement-defined edge occurs (II), or single traversals from both initial nodes each take place (III).

It should be noted that, although this similarity metric is described by the authors in their manuscript on OpenTapioca, the actual similarity metric that is used in the default

configuration according to the code repository follows a more simplified approach.[12]

$$sim_{default}(entity_1, entity_2) = \begin{cases} 2 & entity_1 = entity_2 \vee entity_1 \in l(entity_2) \wedge entity_2 \in l(entity_1) \\ 1 & entity_1 \neq entity_2 \wedge [entity_1 \in l(entity_2) \oplus entity_2 \in l(entity_1)] \\ 0 & otherwise \end{cases}$$

This simplified similarity metric reduces the similarity values to three scores, yielding the highest score 2 if both entities are identical or are bi-directionally connected via their statements, 1 if only a uni-directional direct connectivity between the entities exists, and 0 in all other cases. Figure 3.2 exemplifies the WikiData graph connectivity along the statement-directed edges.

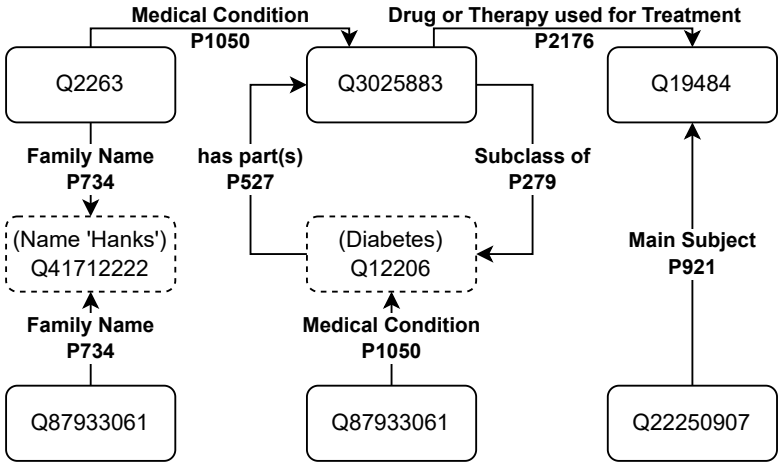

Figure 3.2: Example for the WikiData relationships between the entity candidates in OpenTapioca.

Given the set of detected spans along with their corresponding entity candidates, a new graph is composed of nodes for each pair of span and entity candidate, and each pair of node is connected by a weighted edge, except all nodes pairs that concern a common span or exceed a certain maximum distance between the their span locations. Such a graph is illustrated in Figure 3.3. Conceptually, the edge weights are primarily computed by the similarity metric, but are also influenced by an additional smoothing component as hyperparameter, and furthermore scaled by the distances between the span locations normalized by the maximum distance.

The resulting graph is re-formulated as an adjacency matrix and normalized to a stochastic adjacency matrix. This matrix is used to propagate the local compatibility feature vectors, which are stacked as feature matrix, along the graph structure by performing a matrix-matrix multiplication between the nth exponential of the adjacency matrix and the feature matrix, eventually yielding a final context- and semantic similarity-aware feature matrix.

[12] https://github.com/opentapioca/opentapioca/blob/691b36a89cf7407490472419ceff6846cee0694d/opentapioca/classifier.py#L18 (accessed April 19th, 2024)

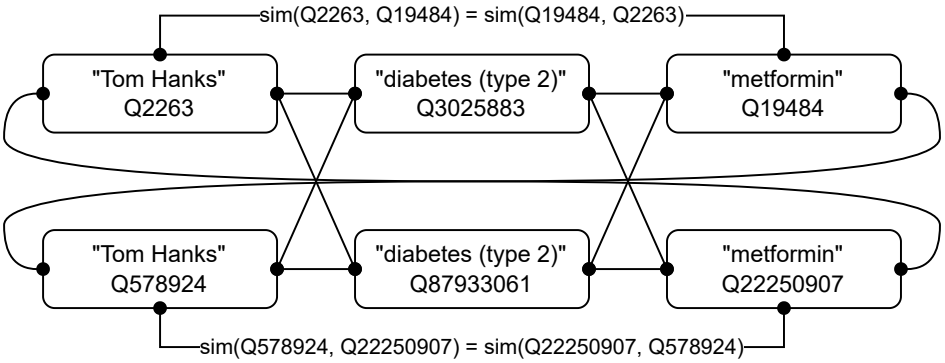

Figure 3.3: Example for a similarity-weighted graph of nodes representing pairs of text spans and entity candidates in OpenTapioca.

Given the number of graph propagation steps m as a hyperparameter, multiple final feature matrices are obtained, given for the adjacency matrices exponentiated between 0 and m-1 each. The final score on the ranking of each entity candidate is estimated by a linear support vector machine (SVM) [123] that is trained on $m \times 5$ normalized input features as the different feature matrices are stacked along the feature axis. The normalization parameters are determined as part of the training and aim to scale and normalize the features to a zero mean and unit variance first.

Hence, in addition to the WikiData resources as well as the Solr- and OpenTapioca-related configuration parameters, OpenTapioca requires the user to provide an annotated dataset for the training of the feature normalizer and SVM classifier.

3.3.3 Spacy

Spacy ("spaCy") [85] is an open-source library for NLP written in Python and Cython. It exposes core API functionality for a simple access to certain NLP algorithms and utilities, as well as adequate linguistic data model schemas, data loading and storing operations and single components that can be composed into larger NLP pipelines.

Spacy features support for multiple languages which span from common European languages like English, French, Spanish or German, but also include functionality for languages of lower popularity such as Finnish or Lithuanian. This language support may be limited to language-dependent information such as vocabulary tables, non-contextualized word vectors or stemming rules, but can also encompass entire pre-trained pipelines of multiple components.

Spacy supports several components that can work on top of a discrete token sequence in combination with a rule-based processing logic, but also components that utilize feature vectors, token embeddings or outputs from other components. For instance, the NER head component does not directly processes the raw sequence of tokens but requires token embeddings predicted by a trained token encoder to eventually predict named entities in unseen texts.

The purpose of the token encoding is to predict or compose a feature vector for each

token. In slim pipeline configurations that are optimized for a compute-efficient pipelines, the *Token2Vec*-based feature vectors are based on hand-engineered features in combination with basic, dense feed-forward neural network layers. Due to the emerged popularity and effectiveness of transformer models like token encoders from the BERT family, Spacy also supports the use of pre-trained transformer models as an alternative option to encode tokens into token feature vectors as dense vector embeddings. In this context, while Spacy uses the internal deep learning library *Thinc*, it also integrates pre-trained transformer models from external sources, such as models from the Huggingface Hub, as token encoders.

Usual pipeline components support several generic NLP tasks, such as part-of-speech tagging (PoS), dependency parsing, stemming/lemmatization, sentence boundary detection or NER. Hereby, the SpaCy offers pre-pretained NLP pipelines for several language both with slim Token2Vec configurations as well as with more compute-intensive transformer-based counterparts.

Prior to the token processing, Spacy offers a tokenization algorithm that follows a rule-based paradigm. The ruleset may be dependent on certain target languages, however the general tokenization strategy in Spacy separates words and punctuation from whitespace characters, but does not further employ subword tokenization. In contrast to other tokenizers, Spacy's tokenization strategy is fully reversible, meaning that a reconstruction of the original text from the tokenized sequence is ensured.

3.3.4 Huggingface Transformers

The *Transformers* library [111] from Huggingface is a Python library that composes basic building blocks to implement a broad range of transformer architectures along with a wide range of utilities and infrastructure that serve common needs in NLP research and applications. In particular, next to different dialects of the variety of transformers it also implements a framework for applying pre-existing architectures with pre-trained transformer weights to common downstream tasks such as text classification, token classification, autoregressive language modeling and masked language modeling, machine translation or text summarization. Beyond the field around NLP, it also features other areas such as computer vision, audio and speech tasks, yet these elements are outside the scope of this work. Concerning the underlying deep learning framework for differential programming, the Transformers library supports three libraries, namely Meta's PyTorch as well as Google's Jax and TensorFlow, as tensor backends.

Within the realm of the Huggingface ecosystem, a key component is the *Huggingface Hub*, a web platform for sharing datasets and trained models. The Huggingface Hub enables an easy and standardized way to exchange data and model weight resources, and thus, it facilitates data sharing and collaboration within the ecosystem. Due to its popularity, the Huggingface Hub interface is not only integrated well into the Transformers library itself, but also used by other NLP libraries like Spacy as mentioned earlier. This enables externally, pre-trained transformer models developed with the

Transformers library to be subsequently re-used for fine-tuning on an NLP downstream task with Spacy without the need for manual model conversions.

3.4 Model Resources

3.4.1 Machine Translation using FairSeq

Prior to the popularity of the Huggingface Hub and its Transformers library, Meta (formerly Facebook) already developed and released the open-source Python library *FairSeq* [124], an ML research toolkit that implements several model architectures from various sequence modeling tasks as well as development scripts around model training and inference. This includes the resources for neural machine translation (NMT) based largely on the vanilla transformer architecture. However, the library can also be used for pre-training a transformer model from scratch, fine-tuning on a downstream task, but also supports non-transformer architectures such as convolutional architectures for sequence-to-sequence models, audio or wave signals frequently used in speech recognition tasks.

FairSeq builds on top of Meta's PyTorch deep learning framework. In this context, PyTorch, in addition to its role as a Python library, provides a platform for model sharing, PyTorch Hub. This platform is closely used by the FairSeq library to facilitate the simple access and exchange of pre-trained models that originate from FairSeq-based pipelines to external users.

In particular, FairSeq provides pre-trained translation models such as the WMT 2019 model [125]

```
transformer.wmt19.en-de.single_model
```

which refers to an English to German NMT model, through the PyTorch Hub.

3.4.2 Bitext Word Alignment

Given a pair of text samples that are mutual translations between two languages, bitext word alignment is the task to establish an alignment between these two text samples on the level of their individual words, such that for a pair and its mappings, a word in the source sentence is mapped to its semantically corresponding words in the target sentence. This means that the mappings are not necessarily bijective.

A statistical approach to model this task originates from research on statistical machine translation. The *Fast_Align* [126] method stems from the IBM model 2 of the IBM alignment model family. A core principle which supports most of such statistical approaches is the assumption that for a set of sentence pairs certain structural commonalities related to word alignments are shared across the sentence pairs. Due to the comparatively computational simplicity of the applied modeling, this model can be used to process large amounts of texts without the need for a dedicated GPU acceleration.

In recent years, the development of neural language models also impacted the methods applied in word alignment tasks. Especially multilingual and cross-lingual models that must be able to process text from multiple languages during the encoding of tokens into embeddings are an essential component for alternative word alignments such as SimAlign [127] and AwesomeAlign [128]. SimAlign directly exploits the property of multilingual models to organize the embeddings space to map identical words and concepts to similar embedding vectors, with the assumption that corresponding words between the sentence pair are encoded into similar embedding vectors. Several decoding strategies for determining the actual word alignments are possible, yet the method itself does not require a parallel corpus nor fine-tuning since it relies on pre-trained transformer-based encoders XLM-R [129] and mBERT [96]. AwesomeAlign follows the principal idea of using multilingual transformers and utilizing their embedding vectors. However, AwesomeAlign combines the fine-tuning on classical language modeling objective with additional objective that contribute towards improving the similarity between embedding vectors of corresponding words. The authors of AwesomeAlign provide such fine-tuned models for public access for several language pairs, including German-English.

3.4.3 Pre-trained Language Models

Several pre-trained language models are used within the context of this work, or are closely related to certain resources involved. These models are presented in this section.

Monolingual, Generic Language Models

The following items concern models that have specifically been pre-trained with strong priority or even exclusively on German text. Therefore, they can be optimized well with regard to the processing of German-only texts over Google's vanilla multilingual mBERT model, including the use of a language-specific tokenization vocabulary. No domain-specific pre-training has been applied, meaning that the use case of these models remains conceptually generic.

German BERT The *German BERT* model[13] is a masked language model developed by Deepset AI released in 2019. The models use the BERT architecture and has been trained on text dumps from the German Wikipedia (6GB), OpenLegalData[14] (2.4GB), and unspecified news articles (3.6 GB). Albeit not explicitly declared by the authors, the model and its tokenizer appear to be initialized from scratch due to its incompatible vocabulary of the tokenizer to other pre-trained models. The final model is 110M parameters in size.

GBERT The updated iteration of *German BERT* models from Deepset AI includes two models, *GBERT* (BERT-based) and *GELECTRA* (ELECTRA-base), both available

[13] https://www.deepset.ai/german-bert (accessed May 7th, 2024)

[14] http://openlegaldata.io/research/2019/02/19/court-decision-dataset.html (accessed May 7th, 2024)

as base and large models [130]. The dataset has been updated to include OPUS[15] (10GB), a collection of various parallel corpora including text from movie subtitles or software po/gettext translation data, and German OSCAR (145GB), a monolingual German subset of the Common Crawl corpus with focus on text quality, in addition to German Wikipedia (6GB) and OpenLegalData (2.4GB). The vocabulary of the tokenizer is identical to a German BERT model[16] developed by Deepset AI and DBMDZ. The GBERT models are 111M (base) and 337M (large) in size, comparable to the GELECTRA models (110M/base, 336M/large).

GottBERT As the first public monolingual model trained on the German subset of OSCAR (145GB), *GottBERT* [131] is another German pre-trained masked language model. For the training, the first version of OSCAR with applied deduplication was used according to its authors. Instead of following the vanilla BERT architecture, it is based on the modified RoBERTa schema, and thus, does not include a next sentence prediction task in addition to the masked language modeling task, and the tokenizer is initialized from scratch. The model is accessible as a base version, being 127M parameters in size.

Medical Domain-specific Monolingual Language Models

Going beyond the training of monolingual language models, the aim of domain-specific pre-training is to optimize a language model to fit well to the linguistic features and patterns of the domain and thus, is expected to improve the downstream performance within a certain domain.

German MedBERT The *German MedBERT*[17] model constitutes one of the first public attempts to perform a domain-specific pre-training on medical-related German text. Yet, the authors of the work, which resulted from a Master's thesis, refer to it as fine-tuning on domain-specific data for the language modeling task since the model is initialized with the model weights of the *German BERT* model. Hence, also the tokenizer is kept identical to the *German BERT* model, implying that no advantage of potentially more effective tokenization is taken. The corpus is based on non-technical summaries (NTS) on Animal Experiments (24MB) as evaluation data, and crawled text articles from German web pages on medical subjects (sprechzimmer.ch[18], netdoktor.de[19], doktorweigl.de[20], onmeda.de[21] krank.de[22] internisten-im-netz.de[23],

[15] http://opus.nlpl.eu (accessed May 7th, 2024)
[16] https://huggingface.co/dbmdz/bert-base-german-cased (accessed May 9th, 2024)
[17] https://huggingface.co/smanjil/German-MedBERT (accessed May 8th, 2024)
[18] https://www.sprechzimmer.ch/ (accessed May 8th, 2024)
[19] https://www.netdoktor.de/ (accessed May 8th, 2024)
[20] https://www.doktorweigl.de/ (accessed May 8th, 2024)
[21] https://www.onmeda.de/ (accessed May 8th, 2024)
[22] https://krank.de/ (accessed May 8th, 2024)
[23] https://www.internisten-im-netz.de/ (accessed May 8th, 2024)

apotheken-umschau.de[24], vitanet.de[25]) (127MB, or 68MB of single sentences) as training data. Since the model is essentially the *German BERT* model with additional pre-training, its size is identical to the size of *German BERT* (110M).

medBERT.de Another model striving towards medical-specific pre-training in German text is *GerMedBERT*'s *medBERT.de* [63]. The model is based on the BERT architecture, but is initialized from scratch along with its tokenizer. The training corpus comprises several sources, including the first iteration of the GGPOnc corpus (10MB) and the crawled dataset from *German MedBERT*, but also includes open data such as Wikipedia, proprietary data from scientific articles and internal EHR data, summing up to 10GB in total training data. While the authors tried to only submit anonymized EHR texts to the training process in order to avoid privacy-related concerns, the access to the model requires the users to agree to its use agreement. In total, the model consists of 109M trainable parameters.

Other Language Models

The previous language models have in common that they are encoder-only models, since they implement the BERT-like masked language model architecture design. Moreover, they focus on German text due to their monolingual training data. While there are notable differences in the size of training data, there are only minor differences in the number of model parameters considering the "base" models, as such models tend to hover around the 100M parameters mark.

GPT NeoX The GPT NeoX 20B [101] is a pre-trained language model from Eleuther AI, that rather relies on the decoder-only architecture typically used for language generation tasks. Another major difference to previous models concerns the model size of 20B parameters. Therefore, it can be rather considered an LLM-sized model and may be used as such, meaning that the model outputs are rather controlled by a given input prompt than by fine-tuning the model itself. Due to its size, the assumption is that global knowledge including all necessary information has already been seen by the language model during the training, given that the GPT NeoX model was trained on *The Pile* [132], a large-scale dataset of various text data, such as scientific texts, web scrapes, e-book corpora and other resources available from the web, summing up to 825GB in total size.

3.5 Implementations of Neural Named Entity Recognition

Prior to the training of a neural NER model, a particular architecture must be defined that is capable of detecting relevant entities from a text. Because there is no unique way of tackling this task, actual implementations differ in several aspects, such as in

[24] https://www.apotheken-umschau.de/ (accessed May 8th, 2024)

[25] https://web.archive.org/web/20231209141110/https://www.vitanet.de/ (accessed May 8th, 2024)

terms of their feature encoding, entity detection, and sequence encoding. Two different approaches are used within the context of this work.

3.5.1 Named Entity Recognition in Huggingface Transformers

Architecture Design

The Transformers library implements a set of generic but task-specific architectural adaptions well suited for fine-tuning on a pre-trained encoder network with an additional task-specific network head. For NER, the architecture employs a token classification architecture identical to Figure 2.5. The encoder network usually resembles a transformer-based model like BERT or RoBERTa. It starts with the initial embedding layers and eventually generates contextualized token embeddings via the attention-based layers that are subsequently used by the additional network head and is therefore called the *base model* within the Transformers library. The NER head resembles a single linear layer which is responsible for the prediction of the tag scores. The structure of the encoder and NER head is shown in Figure 3.4.

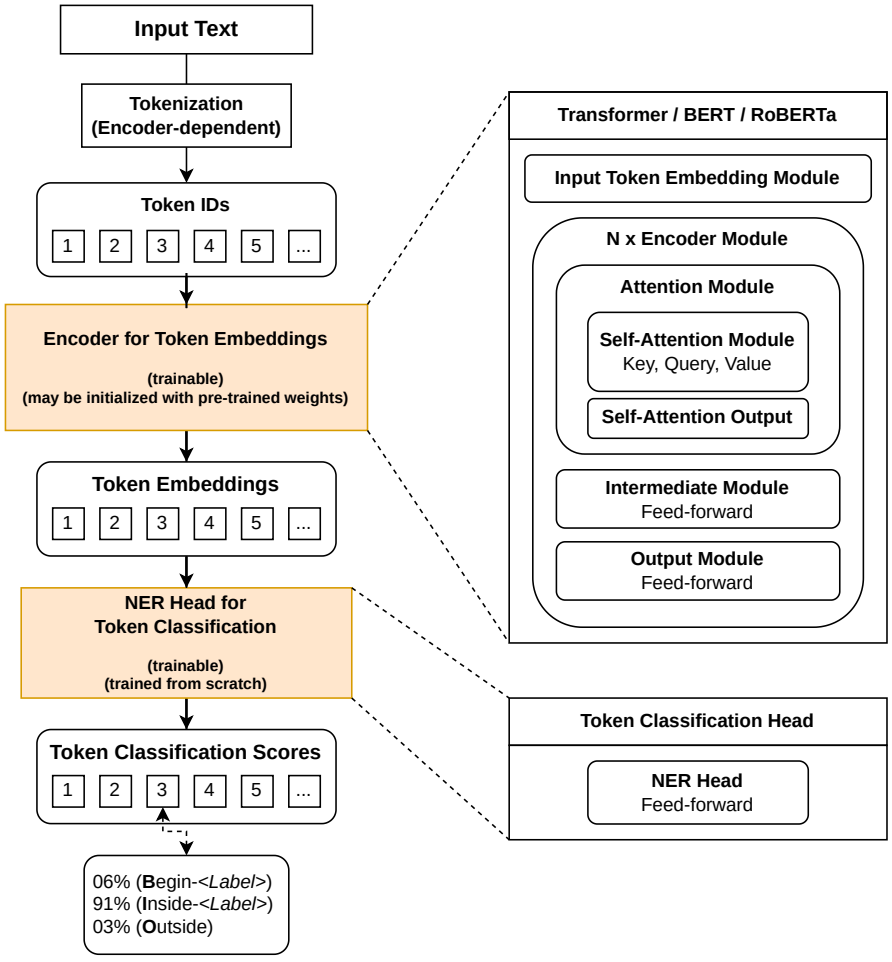

Figure 3.4: BERT-based encoder and NER head components in Huggingface Transformers.

NER Sequence Decoding

By the design of the Transformers library, the implementation for token classification tasks remains NER-agnostic in general. NER-specific modifications are rather absorbed by the use of a pre-tokenized and pre-labeled dataset that has already been transformed according to a certain tagging scheme and these tags are eventually fed to the model as final token classification classes. By this approach, the NER head directly predicts the scores for a token tag given the embedding vector of a certain token. In this step, no other token embeddings are taken actively into account, hence, such contextualized token embedding already includes relevant information from neighboring tokens within the transformer-based encoding stage. The prediction of the tag scores is therefore implemented as a stateless process.

Within this work, three distinct decisions are additionally made when the Transformers library is used for NER. First, the IOB2 tagging scheme is used to encode a labeled sequence for NER. Second, the sequence decoding utilizes a structured prediction approach to avoid invalid sequences during the tag sampling. Third, the NER tags are sampled using greedy sampling. These decisions result in a customized decoding logic which is implemented on top of the Transformers library for this work.

The entire sequence decoding process is shown in Figure 3.5.

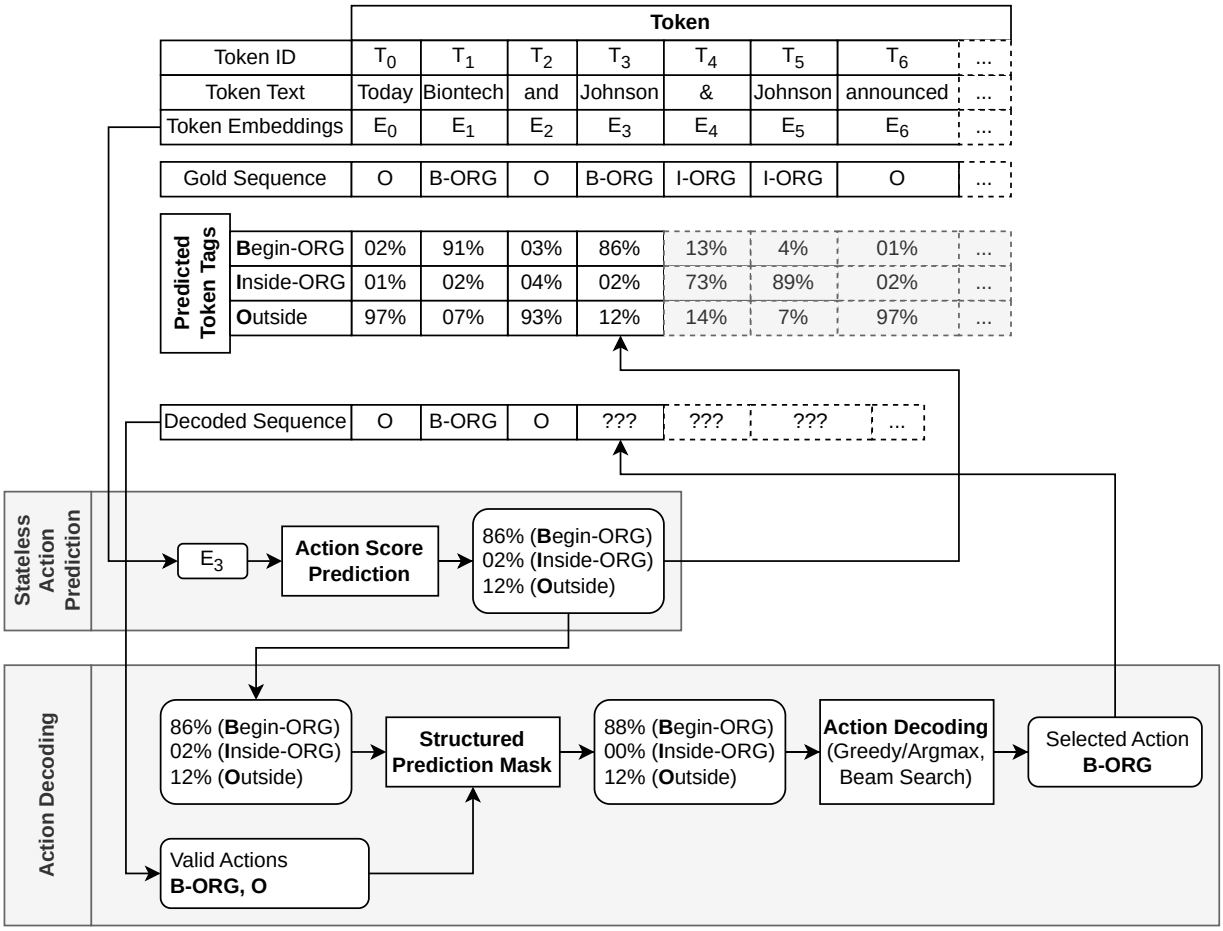

Figure 3.5: NER sequence decoding using Huggingface Transformers.

3.5.2 Named Entity Recognition in Spacy

Architecture Design

Conceptually, the NER architecture in Spacy follows the similar, generic structure as shown in Figure 2.5. As encoder component, the two following options are supported.

- **Token2Vec** (slim): The Token2Vec component extracts certain hand-crafted features from individual tokens, such as their orthographic appearances (identical to the text of the token) or their shape structure which indicate lower and upper case, numeric and punctuation characters. The extracted features are transformed into vectors using *Multi-hash embeddings* [133], in which the extracted features are hashed into integers and mapped to entries within fixed-size embedding tables using the modulo operator. The vectors from the embedding table are further processed by a chain of one-dimensional convolutions with maxout operations and residual connections. Due to this rather slim encoder technique, the computational demand remains rather low and viable especially for CPU-based training. Historically, this approach predates transformer-based approaches.
- **Transformer**: As an alternative to Token2Vec, Spacy provides a wrapper for Huggingface's Transformers that can load and fine-tune pre-trained models from the Huggingface Hub. Since Spacy's tokenization differs from the subword-oriented tokenization of individual transformer models, a second tokenization is required.

The NER head component in Spacy implements a more complex, state-based approach, and hence, is occasionally referred to as a *transition-based NER parser*. When a transformer-based model is used via the Huggingface wrapper, the generated token enbeddings are re-aligned onto the Spacy token structure by using average pooling for multiple subword tokens that are aligned onto a single Spacy token. The obtained vectors are first resized to *hidden* vector representations with a user-defined dimension by a linear layer. Since Spacy adopts the state-based parsing approach, the resized dense vectors are aggregated depending on the current state of the NER parsing, as explained in the next subsection. Given these aggregated vectors, the classification scores are predicted either by a direct feed-forward layer, or a feed-forward layer assisted by a second feed-forward layer which is added if a Token2Vec encoder is applied. The components are illustrated in Figure 3.6.

NER Sequence Decoding

In Spacy, the implementation of the decoding of the label sequence is already predefined as part of the default NER pipeline in large parts. As such, the tagging scheme BILUO is selected. However, a central, distinctive design choice concerns the state-dependent NER parser. Rather than predicting the NER tag scores directly from each token embedding independently, the tag prediction for a token is using the embedding vectors of three token. The first token refers to the current token at which the NER tag scores are unknown so far. To inject further information to the prediction process,

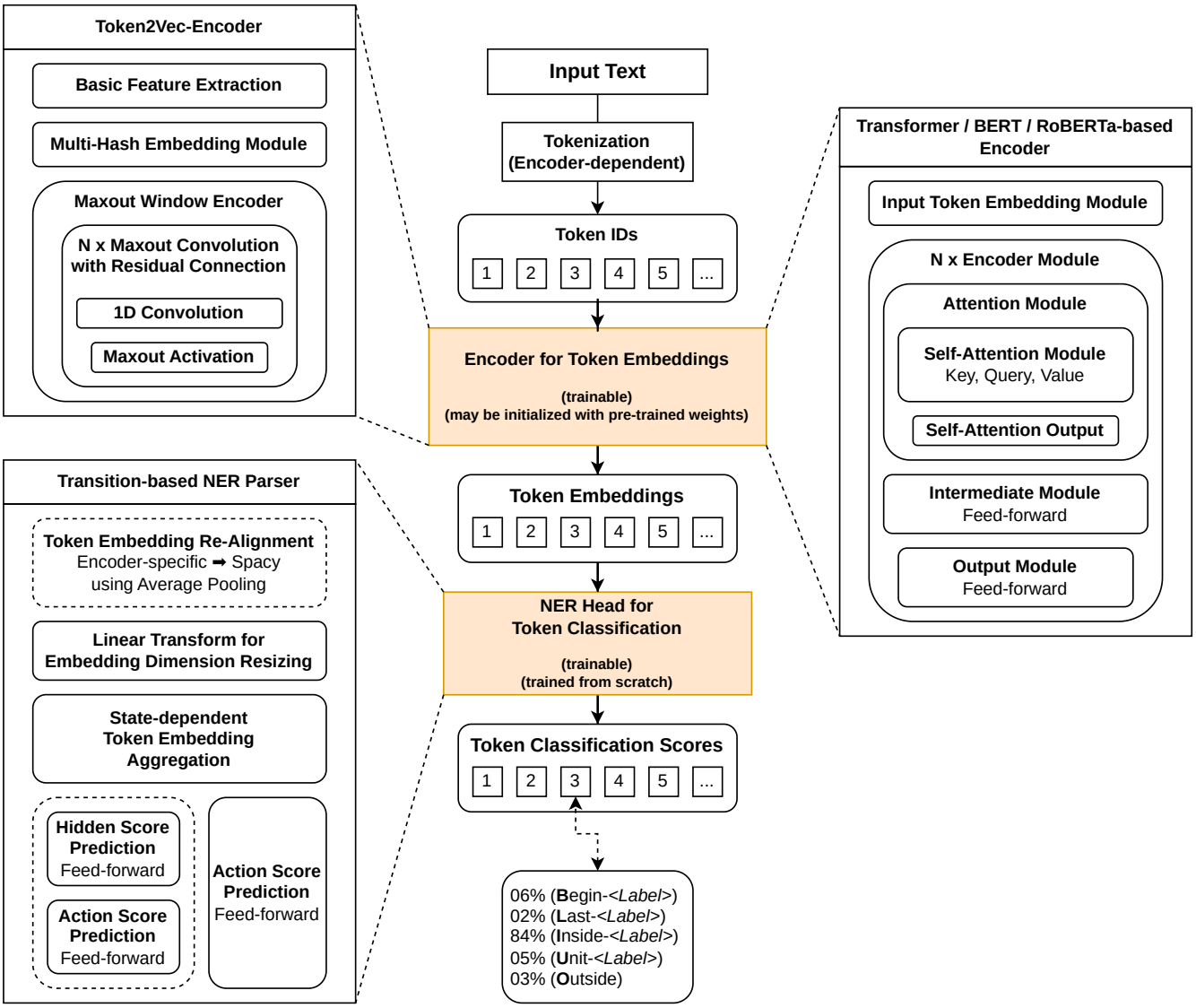

Figure 3.6: Token2Vec-based and BERT-based encoder and NER head components used in Spacy.

two more embedding vectors are added if the previously decoded sequence up to the current position indicates that the parsing state is at the beginning or inside of an entity. In this case, the embedding vectors of the first token of such entity, as well as the most recently decoded token, which describes the token prior to the current token, are aggregated into a large, state-dependent vector. Similar to the decoding strategy in Huggingface Transformers, the sampling from the NER label scores uses an structured prediction approach and argmax to obtain the final NER tag. The overall sequence decoding process is shown in Figure 3.7.

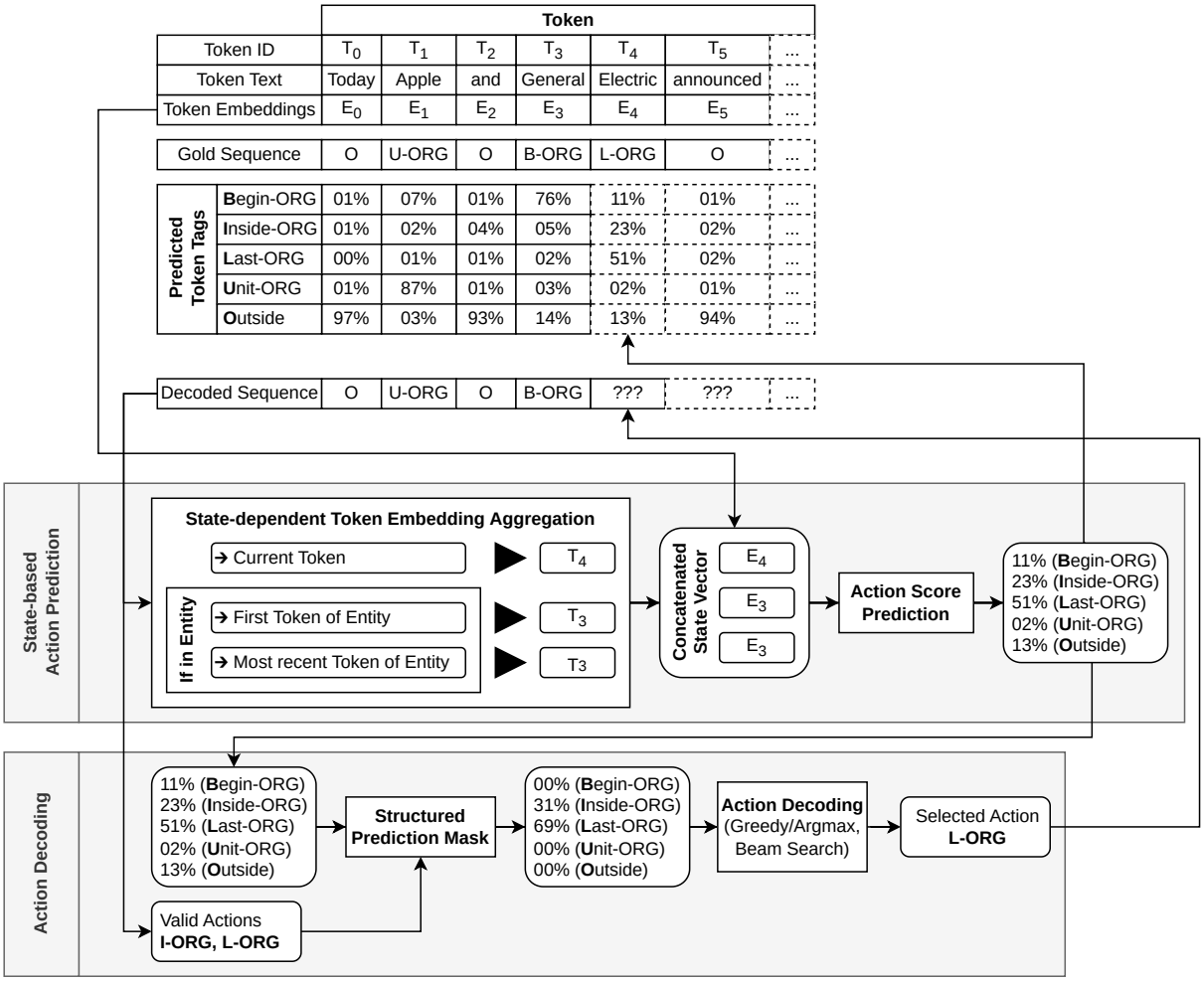

Figure 3.7: NER sequence decoding in Spacy.

Chapter 4

Methods

4.1 Entity Linking via Knowledge Base-grounded Search

An obvious way to detect relevant information and entities in unstructured text is to search the text for specific terms from a pre-compiled vocabulary. For such a vocabulary, entries from UMLS or SNOMED CT are potentially well-curated sources. However, other challenges may render them less suited as match terms, depending on the actual use case. These factors may include:

- **Synonyms and Discrepancies in Style**: Formal definitions of concepts may use a specific, standardized style of language that differs from the terminology used in everyday practice, where more informal or layman's terms are often preferred as synonyms. A dictionary-based matching approach therefore is required to cover alternative terms as well in order to avoid penalties on the recall performance. Although SNOMED CT, for instance, also maintains a list of synonyms attached to each concept, it does not consistently cover layman terms.
- **Lack of Language-specific Terminology**: While UMLS consists of terms from multiple sources that cover German next to English, French, and Spanish, other terminology systems like SNOMED CT lack or only have incomplete language-specific translations.

In principle, several potential benefits of such search-based approaches can be highlighted.

- **Simplicity**: Unlike purely learning-based techniques like neural networks, a terminology-guided search-based approach does not necessarily require annotated and large datasets, and no complex training procedures, and thus tremendously facilitates its practicability, particularly in low-resource domains.
- **Explainability**: The mechanism of the decision on why a particular term has been detected in a text can be directly tracked and explained by the developer, and thus provides direct feedback through its explainability as well as it facilitates the predictability of the behavior on unseen documents.
- **Controllability**: As a consequence of the previous points, the behavior of the

approach can be controlled, e.g. by adding missing terms to, or removing incorrect terms from the vocabulary to increase the overall accuracy.

Nevertheless, due to the success of neural models in NLP, less attention has been given to traditional search-based approaches. However, such implementations serve as baseline results that enable the assessment of alternative approaches.

To this end, the pipeline of OpenTapioca and Apache Solr serves as the baseline approach for the use case of detecting medical entities in text. In order to apply the pipeline, several preparation steps are required. First, the relevant WikiData entities needs to be selected by creating an OpenTapioca *profile*. Second, the WikiData entities need to be loaded and indexed by the Solr collection. Third, an annotated dataset is required to train the SVM classifier from OpenTapioca.

This approach is has been published as scientific work [134]. This approach is referred to as *DrNote* for brevity.

4.1.1 Profile Definition

An OpenTapioca profile is a JSON-encoded definition file that describes which WikiData entities should be added to the Solr collection index to be later taken into consideration for the EL by OpenTapioca. The schema allows the user to select an entity by the sole presence of a certain statement, or by the assignment of a statement tied to a certain target entity. An entity is added to the index if at least one selection criterion is satisfied. Furthermore, additional alias values that may be used to refer to the entity can be defined so they can be covered later during the term indexing stage. This allows additional identifiers, such as ICD codes, to be indexed. A generic example of such profile, a profile for detecting entities of humans, organizations (science-related) and locations, is provided by OpenTapioca[1]:

[1] https://github.com/opentapioca/opentapioca/blob/691b36a89cf7407490472419ceff6846cee0694d/profiles/human_organization_location.json (accessed May 13th, 2024)

```
{
    "language": "en",
    "name": "human_organization_location",
    "solrconfig": "tapioca",
    "restrict_properties": ["P2427", "P1566", "P496"],
    "restrict_types": [
        {"type": "Q43229", "property": "P31"},
        {"type": "Q618123", "property": "P31"},
        {"type": "Q5", "property": "P31"}
    ],
    "alias_properties": [
        {"property": "P496", "prefix": null},
        {"property": "P4550", "prefix": null}
    ]
}
```

In the example profile, every entity is selected that has at least one property satisfied.

- The entity has the statement "Grid ID" (P2427) assigned. A Grid ID is a deprecated identifier for research institutes.
- The entity has the statement "GeoNames ID" (P1566) assigned. A GeoNames ID is an identifier for geographical points of interest.
- The entity has the statement "ORCID ID" (P496) assigned. An ORCID ID identifies a researcher.
- The entity has an "instance of" statement (P31) to "Organization" (Q43229) assigned.
- The entity has an "instance of" statement (P31) to "geographical feature" (Q618123) assigned.
- The entity has an "instance of" statement (P31) to "Human" (Q5) assigned.

As for the indexing of additional alias values, ORCID and CNRS ID (P4550, a French system to identify research organizations) values are included. It should be noted that the entries for "language", "name", and "solrconfig" do not affect the outcome of the pipeline in OpenTapioca. For instance, the language parameter determines the language of the main label indexed by Solr, however the labels and alias values of all other languages are indexed as alias labels as well, resulting in the fact that OpenTapioca remains a multilingual EL system.

To adapt the definition profile to the medical context, the profile is updated as follows.

```
{
    "language": "de",
    "name": "medical",
    "solrconfig": "tapioca",
    "restrict_properties": [
        "P699", "P2892", "P486", "P672"
    ],
    "restrict_types": [
        {"type": "Q12140", "property": "P279"}
    ],
    "alias_properties": [ ]
}
```

Here, the entity selection logic is changed accordingly.

- All entities with a "Disease Ontology ID" (P699), "UMLS CUI" (P2892), "MeSH descriptor ID" (P486), or "MeSH tree code" (P672) statement are selected.
- All entities with a "subclass of" (P279) statement to "medication" (Q12140) are selected.

This configuration aims to cover a majority of key vocabulary in the medical context. Even though the entity selection may cover more entities than intended for the EL task in certain use cases, for instance in case of medication detection which does not require any disease-related entities, such entities can be filtered out in a post-processing step.

4.1.2 Data Preparation

Independently of the profile definition, two statistical items must be computed for the EL step to function. The first item concerns the creation of an unigram language model, which is used to estimate the prior probability of a certain text span. This is established by collecting all *English* label and alias phrases from the WikiData stack. The fact that only English terms are used throughout this process remains a limitation of the current OpenTapioca implementation. The phrases are normalized to ASCII symbols and split along whitespaces into words, added to the language model and its count incremented accordingly. The second item, the computation of the PageRank vector of all entities, is computed likewise on the WikiData structure. To facilitate the computation on the graph, represented as large adjacency matrices, the operations are implemented using sparse matrix representations.

4.1.3 NIF Synthesis using Wikipedia

The last challenge concerns the training of the SVM classifier and requires a dataset with annotated entities. Because of OpenTapioca's EL task on WikiData, the annotations must reference to WikiData entities specifically as well. To avoid the need for the manual input to provide with such an annotated text corpus, the objective is to

synthesize the dataset from the existing data sources, Wikipedia and WikiData, in an automated fashion.

Concept for Annotated Text Acquisition A language-dependent Wikipedia snapshot dump consists of a set of pages, composing conceptually both the WikiText-encoded page text and its unique page title. Within the WikiText, certain phrases might be annotated to point to other article pages within the same Wikipedia language scope, commonly rendered as hyperlinks in web browsers. Re-purposing phrases that reference other pages as annotated entities is the driving idea behind the creation of a fully automated corpus. However, only phrases that are actually annotated in WikiText can be converted to annotation data, yet potential phrases frequently remain unannotated. For instance, only the first mention of a certain phrase referring to another Wikipedia page is usually annotated within the context of a Wikipedia page, whereas the latter mentions of that phrase are left unannotated. Therefore, the corpus creation only yields *sparsely*-annotated corpora, also is also referred to as *weakly*-annotated corpora in literature. It remains an inherent drawback of utilizing Wikipedia-based annotations.

So far, when considering Wikipedia as a network of interconnected Wikipedia pages, its structural accessibility is limited by the lack of a coherent hierarchical structure which is a key property of ontologies that are intended to organize knowledge in structured concepts. To this end, WikiData is a knowledge base that features such ontology-inspired structure, and moreover for that reason it can be technically leveraged by the SPARQL query language to query certain knowledge base concepts. In particular, individual concepts, namely WikiData entities, can be filtered and aggregated into a query set match by a certain SPARQL query.

To combine the Wikipedia and WikiData data sources, corresponding links between individual WikiData entities and Wikipedia pages need to be established. For WikiData entities, an entity may include so called *sitelinks* to link an entity to certain language-specific Wikipedia pages. While only one Wikipedia page can be referenced for each language at most for a WikiData entity, it is possible that multiple entities reference an identical Wikipedia page. In particular, some instances of Wikipedia pages are redirection pages, that solely encode a redirection to another Wikipedia page. For instance at the time of writing, *type 2 diabetes* (Q3025883) links to the page *Typ-2-Diabetes* of the German Wikipedia, which is subsequently a redirect to the final *Diabetes mellitus* page, whereas *diabetes* (Q12206) directly links to the *Diabetes mellitus* page of the German Wikipedia. Hence, establishing distinct, bijective links between WikiData and Wikipedia is not always feasible under certain conditions, but this only occurs rather rarely[2].

The concept of the overall process is shown in Figure 4.1.

[2]Given a German Wikipedia dump from February 22, 2024 and WikiData dump from February 23, 2024, only 23 Wikipedia pages were directly referenced by more than one WikiData entity, and 17,830 pages were referenced by multiple WikiData entities through redirection pages. In total, 2,873,496 German Wikipedia pages could be linked to WikiData entities, only 4,391 pages were not linkable and consist mostly of misspelled page titles.

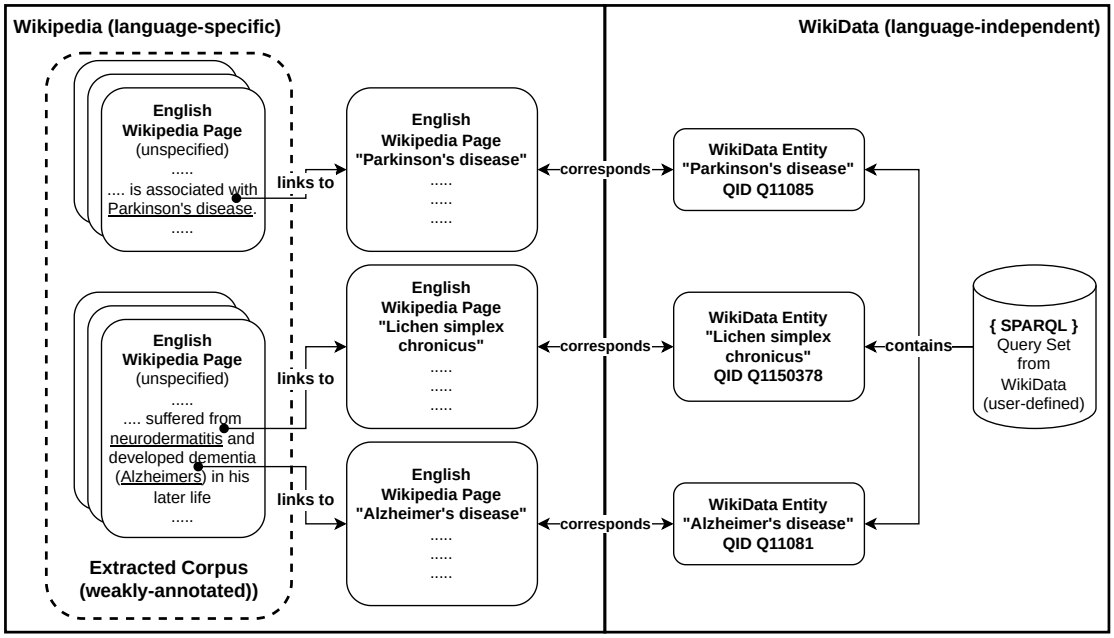

Figure 4.1: A conceptual illustration of the extraction of annotated texts from the WikiData and Wikipedia data sources.

Identification of WikiData Entities using SPARQL To obtain a corpus with annotation data that covers all WikiData entities that have also been indexed by the OpenTapioca EL system, these same WikiData entities must be selected for the corpus acquisition. Because only one of the property-related conditions need to be met, an OpenTapioca profile definition can be transformed into a corresponding SPARQL query by expressing the unions of condition queries using the *UNION* operation as follows.

```
SELECT ?item WHERE
{
 # items from restrict_properties
 { ?item wdt:<restrict_properties.0> ?something }
 UNION
 { ?item wdt:<restrict_properties.1> ?something }
 ....
 UNION
 # items from restrict_types
 { ?item wdt:<restrict_types.0.property> wd:<restrict_types.0.type> }
 UNION
 { ?item wdt:<restrict_types.1.property> wd:<restrict_types.1.type> }
 UNION
 ....
}
```

The texts within the brackets < and > indicate the placeholders for actual data values from an OpenTapioca profile definition.

Synthesis of NIF-encoded corpus OpenTapioca requires a training corpus to be encoded in the *NLP Interchange Format* (NIF). A file in NIF format essentially stores a raw text along with the annotated references links to individual target items, such as Wikipedia pages or WikiData entities. Since OpenTapioca operates only on WikiData entities, the NIF file contains the raw text from the extracted corpus and its weakly-annotated annotation data linking to the individual WikiData entities.

Regarding the implementation details, the WikiText is decoded using the *wtf_wikipedia*[3] parser which transforms the WikiText into a hierarchical representation that follows the Wikipedia-specific, nested block structures. Due to certain historic implementation details, only a fixed block structure is supported by the text extraction pipeline, and more complex and convoluted block structures that do not follow the expected page structure are skipped in this work for this approach. Furthermore, to extract only sentences with relevant annotations, the text blocks from the parser are first merged into a single text sequence and re-split into individual sentences using the Spacy sentencizer.

4.1.4 Classification with Weak Annotation Constraint

The synthesized NIF file is used to train the SVM-based classifier. In this context, two limiting factors may impede the convergence and subsequently the effectiveness of the entire EL pipeline. First, the classifier is trained on a weakly-annotated corpus which incentivizes the classifier towards a low recall rate on actual data, because certain phrases in the training corpus may be left unannotated, although they belong to the actual annotation manifold. Second, the Solr-based entity detection implements a search-based approach on a fixed set of terms and therefore is unable to detect terms beyond its predefined vocabulary. Consequently, certain phrases, e.g. un-indexed synonyms to indexed terms, cannot be detected by training the classifier, eventually leading to an additional loss in recall performance across the entire EL pipeline.

To mitigate the described limitations in a simple fashion, the EL pipeline of OpenTapioca is modified to propagate the set of entity candidates for a detected text span further even if the classification scores of the candidates do not exceed the threshold to consider one of the candidates to be a match to the text phrases. It effectively disengages the ability of the EL system to reject the linking of certain text spans.

4.1.5 Linguistic Filtering

While the former action aims to increase recall scores, this potentially comes at the expense of the precision scores. To compensate for this fact, additional filters as post-processing steps are applied that can incorporate certain assumptions about correctly linked entities. In practice, basic features such as span lengths as well as language-dependent, linguistic features are utilized to filter incorrect linked entities. Concerning linguistic features, it is assumed that all tokens within a detected span must not be part

[3] https://github.com/spencermountain/wtf_wikipedia (accessed December 21st, 2024)

of the set of *stopwords*, a language-dependent subset of words that do not carry any firm semantic meaning. Moreover, it is assumed that at least one token from a detected span is tagged by a language-dependent part-of-speech tagger as a *noun* or *proper noun*. In order to implement these post-processing steps, the pre-trained pipelines of the Spacy library are leveraged which are provided by Spacy and support dedicated languages.

4.2 Named Entity Recognition using Translation and Annotation Projection

In the last decade, tremendous advances in the field of neural machine translation (NMT) have been accomplished, and thus, are also raising the question on whether the increased quality in automatic translation can be re-purposed to convey resources from other languages to the language of a certain target domain. In particular, several and differently severe resource issues could motivate this question in practice. First, no adequate text and annotation resources exist for the target language at all. Second, pure text resources may exist but lack any annotation layer. Third, text resources with some annotation information exist but certain desired annotation data relevant for a specific context are missing.

This study investigates NER as the target NLP task, implying that the challenge of translating the training data encompasses not only plain text but also the corresponding label information. However, other NLP tasks may not require all the steps discussed in this section. For instance, obtaining training data for ordinary text classification tasks from other languages can be realized purely by a simple translation of the text sequences because their text classification labels are assumed to be preserved.

For NER, the situation is more complex because each sample of an NER dataset consists of the plain text as well as a list of annotated entities. Each entity is encoded by its individual span position in the text in combination with its associated label class. For obvious reasons, when translating a text sequence from the source language into the target language, the spans of the entities do not align onto the translated text anymore as they are not invariant to translation.

It is worth considering that, while in this work a translation-based approach is investigated for the German target language, the translation quality and the subsequent performance of the obtained model depend on the availability and accuracy of the pre-existing NMT models. Because these factors can vary depending on the source language and target language, especially very low-resource languages, the feasibility of a translation-based approach may need to be assessed on a case-by-case basis.

Translation at Time of Training versus Inference In the broader context of applying NMT as a tool to leverage resources from other languages, two paradigms could be pursued. When training resources or implementations for a specific NLP task are available in the source language, translating the input text into that language can serve as a preprocessing step that enables the direct re-use of these pre-existing resources. This concept to use translation at inference aims to bring the input data to the pre-existing NLP models in their source language. However, within the context of this work, an alternative approach is investigated that instead aims to bring the pre-existing materials and resources to the target language. Conceptually, these resources needed for the training of a task-specific NLP model are translated into the target language

and used for the model training. Therefore, the processing of input text can be done without a shift in the language domain.

Depending on the actual NLP task, the final output may be rather language-agnostic, e.g. for ICD code annotation of mentioned concepts, and hence, both paradigms are conceptually applicable. However, processing a translated input text in the source language adds another step to the NLP pipeline which may increase the susceptibility to errors, as well as it may impair the transparency and interpretability, especially when subsequent processing steps such as entity detections are solely based on the translated text, and thus, are further detached from the input text. In contrast, although the development of an NLP model on translated training data may suffer from translation artifacts as well, having such model already in the target domain facilitates the auditability, and the output remains grounded to the original input text. Due to this aspect and because the former paradigm falls rather within the scope of non-German NLP research from a scientific point of view, the former paradigm is not discussed further in this work.

Both described paradigms are illustrated in Figure 4.2.

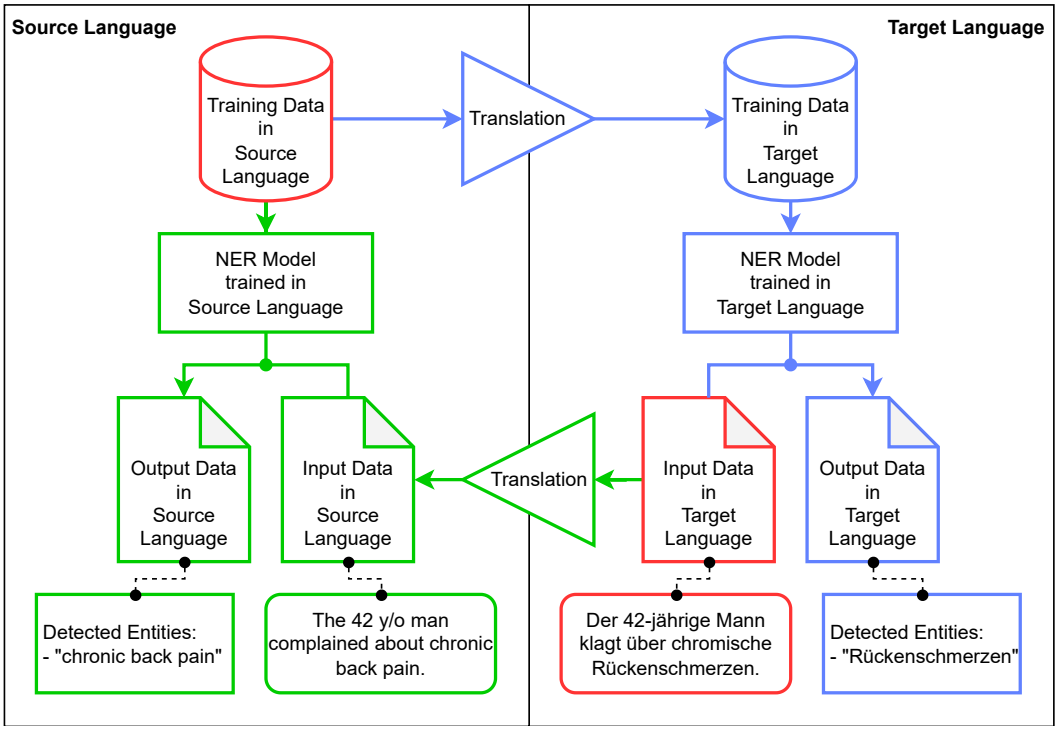

Figure 4.2: Illustration of two translation-based approaches in the context of an NER task. The paradigms are indicated in colors for translation at training time (blue) and translation at inference time (green). The red elements designate pre-existing data resources and input data.

In general, the research question in this section concerns the practicality of developing a German NER model for medical entities based on a non-German dataset by the means of translation, as well as the level of performance that can be achieved.

The proposed concept has been documented in scientific publications and are presented in this section. Two iterations of the approach have been implemented, referred to as the first and second iterations. The first iteration, detailed in [135, 136], is named *GERNERMED*, while the second iteration, described in [137], is named *GERNERMED++*.

4.2.1 Reference Dataset for NER

The dataset from the n2c2 2018 ADE Track 2 challenge serves as an instance of an annotated corpus with entities and relations about medication information and adverse drug events in English, the source language within the context of this study. Of the 505 samples, 303 are published as training data, and 202 are designated as test data, which were held back from participants as a testing resource during the execution of the challenge. The statistics on the raw dataset are given in Table 4.1.

		n2c2 ADE Dataset	
		Training	**Test**
Text	#Chars	3,839,693	2,535,182
	#Tokens	957,906	627,641
	#Sents	42,182	28,206
Entities	#ADE	959	625
	#Dosage	4,221	2,681
	#Drug	16,225	10,575
	#Duration	592	378
	#Form	6,651	4,359
	#Frequency	6,281	4,012
	#Reason	3,855	2,545
	#Route	5,476	3,513
	#Strength	6,691	4,230

Table 4.1: Dataset statistics of the n2c2 ADE dataset. The tokenization was performed using the English tokenizer from Spacy. The sentencizer from Spacy was used for the sentence boundary detection.

For the first approach, only the 303 training data documents are used as input to the translation-based processing. In the second iteration, the 202 test samples are also included to eventually comprise a dataset of 505 annotated documents in total.

4.2.2 Dataset Preprocessing

The raw dataset is organized as a set of documents that are preserved in their coherent structure, meaning that the order of sentences within a document was not changed and the documents were not split into shorter fragments. However, because the raw text data is derived from the MIMIC-III notes, several pseudonymization and masking steps have been applied to protect PHI and other information, such as dates or locations, that could increase the risk of re-identification. For instance, the following text keeps the name of the hospital and a last name of a physician hidden.

```
transfered to [**Hospital1 18**]/[**Doctor Last Name **] for continued
care.
```

Therefore, several preprocessing steps are conducted.

Mask Replacement In order to reduce the occurrences of irregular text artifacts from the pseudonymization-motivated masking, the text corpus is parsed using regular expressions to detect and replace such text masks by adequate surrogate values. The matched patterns include mostly date-related, address-related and name-related (first name, last name) information as well as other identifiers like *medical record number* and *health care provider number*. Replacing these masks by actual values may contribute to a more natural appearance of the text, and thus the step can help to better resemble the actual data samples at inference time. Moreover, the irregular character structure of the mask may lead to an inefficient tokenization of the input text in later stages, raising issues in particular during the translation when only a limited number of tokens can be processed. Regarding the previous text example, the mask-replacement process may result in a line resembling the following.

```
transfered to LMU Grosshadern/Dr.  Meyer for continued care.
```

To sample random, yet plausible values for individual text masks, the Python package *Faker*[4] is used. Because the length of the detected masks does not necessarily match the length of the replacement text, all span-dependent annotation data must be updated during the replacement step, including the start and end character positions of each named entity.

Sentence Extraction Furthermore, the dataset is decomposed into a set of single sentences. This is merely motivated by technical and practical reasons to limit the length of single text samples because document-sized samples may easily exceed certain limits such as the number of maximum allowed tokens when processed by common BERT-like encoders. It may also prevent outlier sentences from triggering negative effects to other sentences within the same sample in later processing stages, such as during sentence translation. The English sentencizer from Spacy is used for the sentence boundary detection. During a preliminary investigation, several extracted samples are found to not follow typical sentence structures and appear like remnants from the header or footer sections, or may be completely unrelated to medication instructions and related entities. Therefore, the set of sentence samples are filtered further to only be retained as part of the dataset if at least one annotated entity is part of the sentence.

4.2.3 Text Translation

The text samples from the dataset are translated given the set of single sentences. The WMT'19 translation model [125] for NMT is applied, which implements a transformer model trained to translate text primarily from English into German. For practical reasons, to ensure that no technical limits are exceeded in terms of input length as well as output length, all sentences that are longer than 1024 characters are disregarded for

[4] https://github.com/joke2k/faker (accessed May 31th, 2024)

translation and removed. After the translation, a parallel corpus consisting of pairs of English and German text samples is generated. However, the annotation data for the named entities is available only for the English sentences.

4.2.4 Annotation Projection

The annotated entities need to be reconstructed for each translated sentence in the target language. To this end, the semantic correspondence between individual words, or even subwords, of the sentence in the source and target language must be identified to subsequently project span-related information from the input sentence onto the translated sentence. To obtain such correspondence, two pre-existing bitext word alignment techniques are employed, namely *Fast_Align* for the first iteration, and *AwesomeAlign* for the second iteration.

Most word alignment models, including the former models, produce word alignment mappings on a word or token level rather than on character level. Given the parallel corpus of sentences, all sentences must be tokenized before being fed into the word alignment model. For the first iteration, the implementation conducts a tokenization by splitting each sentence at whitespace characters, and therefore avoids any more complex text preprocessing. Hence, the following sentence

```
This is an example sentence, including a comma!
```

is tokenized into the following tokens:

```
"This" "is" "an" "example" "sentence," "including" "a" "comma!"
```

The second iteration improves this tokenization step by utilizing Spacy for language-specific tokenization and punctuation detection. Given the previous example, a tokenization yields the following token sequence:

```
"This" "is" "an" "example" "sentence" "," "including" "a" "comma" "!"
```

The resulting alignment mappings are commonly encoded in the *Pharaoh* [138] format which describes the word alignment of the tokenized input sentence to the tokenized translated sentence. From a technical point of view, the format describes a directed mapping of tokens from the input sequence to the tokens of the translated sentence based on the index position of a token.

In preliminary experiments, the resulting word alignment mappings were found to be susceptible to two main issues. First, the output may produce only sparse mappings, leaving several tokens without corresponding words, which can contribute to an increased false negative rate. Second, sentence-level outlier alignment errors may occur, often exhibiting markedly different and unnatural structures compared to correct alignments, especially when using the *Fast_Align* tool.

Because the first iteration uses *Fast_Align*, an additional postprocessing step is used that implements a heuristic decision boundary to classify a sample-wise alignment output as either valid or invalid. Given the word alignments for a sample encoded as a binary matrix, the heuristic can be interpreted as the mean distances of the non-zero

alignment cells to a diagonal line, starting at the top left cell to the bottom right cell, to detect potential outliers.

When considering the following two potential alignment matrices, $A_{correct} \in \{0,1\}^{w_{de} \times w_{en}}$ as the correct alignment matrix and $A_{flawed} \in \{0,1\}^{w_{de} \times w_{en}}$ as an outlier due to its abnormal, hence invalid, alignment structure

$$A_{correct} = \begin{array}{r} \\ \text{Das} \\ \text{ist} \\ \text{ein} \\ \text{Beispiel} \\ . \end{array} \begin{array}{c} \begin{array}{ccccc} \text{This} & \text{is} & \text{an} & \text{example} & . \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$A_{flawed} = \begin{array}{r} \\ \text{Das} \\ \text{ist} \\ \text{ein} \\ \text{Beispiel} \\ . \end{array} \begin{array}{c} \begin{array}{ccccc} \text{This} & \text{is} & \text{an} & \text{example} & . \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

the decision boundary to detect valid alignments for a given matrix $A \in \{0,1\}^{w_{de} \times w_{en}}$ is defined as follows.

$$\frac{1}{max(w_{en}, w_{de})} \sum_{i=1}^{w_{de}} \sum_{j=1}^{w_{en}} A_{ij} \frac{|w_{en} - i \cdot w_{en} + i - w_{de} + j \cdot w_{de} - j|}{\sqrt{(w_{en}-1)^2 + (w_{de}-1)^2}} > \tau$$

Hereby, w_{lang} refers to the number of tokens of a sentence in a certain language and τ is a threshold hyperparameter. Rephrasing the decision boundary as a conceptual interpretation, the underlying assumption, which governs the heuristic, expresses the notion of alignment locality. Tokens at the very beginning of a sentence are expected to correspond to tokens rather at the beginning and are unlikely to correspond to tokens towards the end of the translated sentence.

The filtering of potentially flawed word alignments using the decision boundary is applied only for the first iteration which implements Fast_Align as word alignment tool. For the second iteration, AwesomeAlign is used for word alignment, which less frequently yielded such outlier-like misalignments in preliminary investigations. However, this effect might not be only due to the underlying method, but may also be attributed to the previously described differences in the tokenization.

To finally obtain the NER annotation for the translated sentences, each span of a named entity is projected onto the translated sentence by the following steps: First, the start and end character positions from the initial span are resolved to the token level such

that the first and the last token affected by the span is identified. Second, the start and end positions of the tokens are projected onto the tokens according to the word alignment and their token positions are identified. Third, the final character-wise span for the given entity is reconstructed by retrieving the start and stop character positions of the identified tokens, effectively reversing the tokenization process conceptually. The reconstructed span along with its label class is treated as the final NER annotation data for the translated sentence. While this entire annotation projection process is rather straightforward, it is applied with several implicit assumptions in mind, for instance, that no disjoint spans will occur. Because the samples from all label classes tend to consist of short encapsulated phrases or even just single words, the problem of handling disjoint spans accordingly is not further addressed. Yet in the case of other datasets with annotated entities severely affected by disjoint spans, the need for additional adjustments may be required.

The overall annotation projection pipeline is illustrated in Figure 4.3.

4.2.5 NER Model Training

To avoid the training of an NER model from scratch, using transfer-learning on pre-trained language models can facilitate the training of a final NER model by only fine-tuning an already pre-trained model on a certain NER task, leveraging the effective token embeddings that have already been acquired during the pre-training stage.

For the first iteration, the efficiency-focussed NER pipeline of Spacy is used which relies on the Token2Vec component for token encoding. The second iteration is both trained on a Token2Vec encoder as well as on a transformer-based encoder in different training sessions.

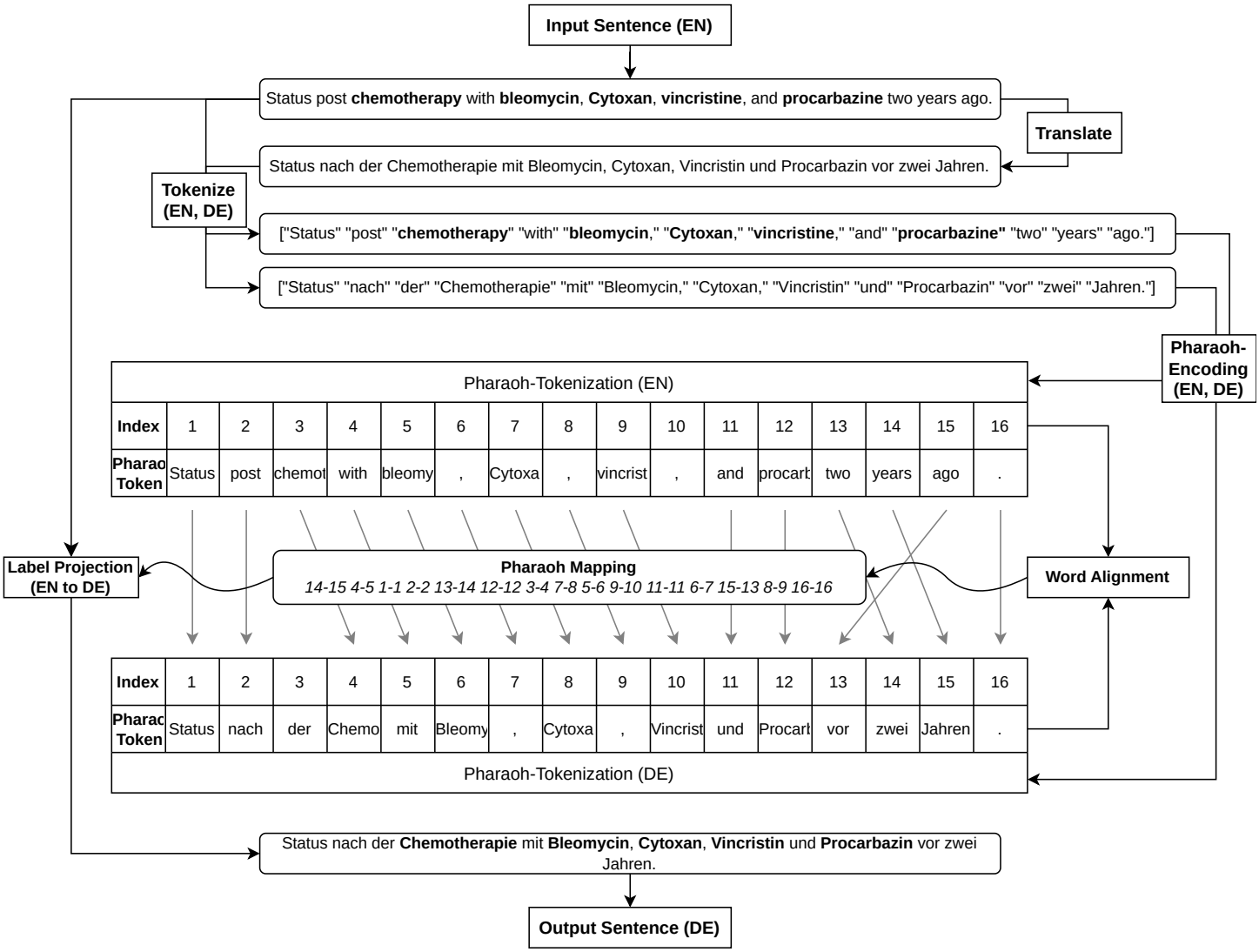

Figure 4.3: Annotation projection using word alignment in the Pharaoh alignment format. The language-specific tokenizers from Spacy are used for the tokenization. In this example, the Pharaoh tokenization treats punctuation tokens as separate tokens. The actual Pharaoh token mapping as well as the text serves only illustration purposes.

4.3 Named Entity Recognition using LLM-generated Annotated Corpus

While modern neural, transformer-based NLP tasks like NER are frequently based on BERT-like encoder-only transformer blocks for tasks like text classification or token classification, autoregressive LMs that resemble decoder-only transformer architectures, are preferred with regard to certain tasks like text generation. Whereas most of the popular BERT-like models publicly available with pre-trained weights, such as GottBERT (base) or German BERT, are consistently less than 1B parameters in size, GPT-like LMs have increased dramatically in size in terms of their number of parameters in recent years. For instance, the smallest GPT-2 model from OpenAI features 137M parameters, however its successive model, the original GPT-3 model from OpenAI is reported to count 175B parameters. While OpenAI has not shared their large-sized models like GPT-3 or GPT-4 openly as model weights but only provide limited access via an API, a broad variety of similar autoregressive models in several sizes have been developed, and with their model weights shared publicly in some instances. As such, GPT NeoX is a publicly available model that features 20B parameters. In order to train such large-scale LMs appropriately, the size and diverseness of the training corpus is highly important as well.

Contrary to the use of encoder-only transformers in which the model is either used as encoder to generate semantically useful hidden representations, or even fine-tuned on a particular downstream task, a simple technique to leverage decoder-only models for certain downstream tasks is to use an input *prompt* which, for instance, includes the problem statement and additional few-shot examples, and is commonly phrased in a consistent syntactical structure. This approach takes advantage of the conditioned text continuation objective to provide the correct solution, given the assumption that the most likely text continuation matches the correct solution to the problem stated in the prompt. Since LLMs are capable of memorizing global knowledge from the large-scale training data, this approach can succeed well even for few-shot or even zero-shot prompt inputs without any further task-specific fine-tuning.

In order to improve the usability, LLMs may be fine-tuned after generic pre-training on *instruction* datasets. This procedure aims to optimize the LLM to adhere to a more consistent prompt style, which consists of an instruction text and may include additional supplementary information or examples, and to a consistent response style, which is frequently framed within a conversational, chat-like context. Furthermore, this chat-like usability facilitates the interaction of LLMs with persons who are inexperienced with the generic text continuation feature of raw LMs.

However, LLMs are computationally expensive, especially during pre-training and fine-tuning, in comparison to more traditional LMs like rather modest-sized BERT models. This aspect even applies to the plain inference without any backpropagation involved. Hence, in low resource compute environments, this may even affect the technical feasibility of deploying LLMs in certain application use cases. Moreover, the

considerable computational complexity and the substantial energy requirements of such a model have implications for its environmental and economical impact, which may render its long-term use unsustainable for the processing of large volumes of data.

In recent years, LLMs have been also evaluated for tasks like NER. Albeit the question of whether LLMs are generally a better fit to medical NER tasks compared to fine-tuned BERT-based NER models is not addressed in the context of this work, this section concerns the question on how domain-specific, implicit knowledge can be extracted from a pre-trained LLM to be distilled into a smaller and thus, computationally more efficient NER model without the need for substantial human efforts. More specifically, the use of an LLM to generate a synthesized dataset with corresponding annotation information for medical NER is investigated. Such synthetic datasets can be used as training data in a follow-up step to train a traditional NER model. Doing so also takes advantage of the fact that such corpus, which is synthesized by a publicly available LLM, is not tainted by sensitive data that may limit open usage and public data sharing, given the case that the LLM in use has not been trained on privacy-related, sensitive data.

The proposed approach, described in the following section, has been published as a scientific work [139] and is referred to as *GPTNERMED* for brevity.

4.3.1 Choice of LLM

The pre-trained GPT NeoX model with 20B parameters serves as the underlying LLM, which is used for text generation purposes. At the time of development, this model was one of the largest LLM models with publicly available pre-trained weights and source code for training and inference, and yet sufficiently small to be loaded to a locally available GPU cluster. While other closed source LLMs with large model sizes and stronger capabilities were available at that time, they were rejected in favor of the open-source GPT NeoX model to ensure basic reproducibility beyond the conceptual level.

4.3.2 Prompt Design using Markup-based Annotation Encoding

Because the GPT NeoX model was trained on the dataset *The Pile* and was not further fine-tuned on instruction-oriented corpora, its primary strength lies in the raw text continuation aspect, which subsequently affects the design of the input prompt. In this context, the key objective for the creation of the synthesized dataset is the acquisition of *annotated* text data. To address this, the input prompt is composed of a set of pre-annotated sentences which were annotated manually with certain label classes and serve as few-shot examples. Hereby, the annotation information is injected into the plain text by using a basic markup encoding which denotes both the start and end of a certain entity as well as its assigned label class. Given an annotated sentence, the text is encapsulated by the opening `<s>` and `</s>` markup tags and indicate the start and end of the text sample. In this case, the tags do not refer to the special *Begin of*

Sentence and *End of Sentence* tokens but are part of the prompt input text. Encoded individual sentences are separated by a new line. Within the text, named entities within the sentence are encoded as spans, denoting the beginning and label class of the span by `<class="label class">`, and its end by `</class>`. The objective of the LLM is to continue the input prompt with more text samples with semantically augmented text and annotation data while preserving the syntactical structure of the markup language. The concept is illustrated in Figure 4.4.

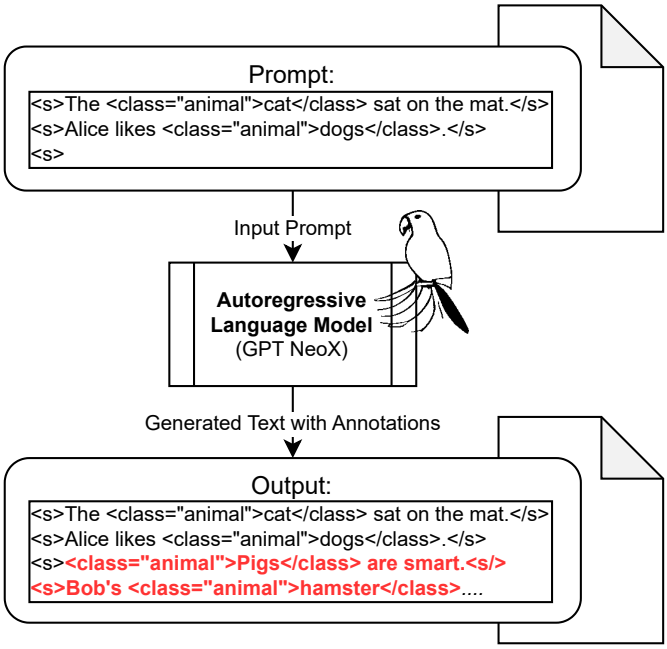

Figure 4.4: Prompt design with markup-encoded text. The named entities are encoded by the markup tags. The synthetical data, indicated as red text, is sampled by the text continuation feature of the GPT NeoX model.

Using this prompt design provides a straightforward way to encode sentences with their annotated entities. The reversed process, namely the decoding of the generated text into a set of named entities, is feasible if the LLM-generated text obeys the implicit rules of the markup structure. Conceptually, this process aims to augment the input examples to further diversify the dataset. In this context, the use of the LLM for the prompt completion represents a simple way to extract knowledge from the LLM weights acquired during pre-training.

4.3.3 Text Sampling using Temperature Parameter

For text generation, the next token prediction is performed by sampling the token from the predicted multinomial probability distribution over the vocabulary of the tokenizer, which is established by applying the softmax function to the predicted logits. In the simplest case, greedy sampling could be applied whereby the token with the highest score is always chosen. However, this method is not optimal for sampling natural text because it is known to produce unnatural artifacts such as repetitive text sequences [140]. Therefore, sampling from the multinomial probability distribution

may be preferred in these cases instead. Additionally, several techniques prior to the token sampling are usually considered.

As previously mentioned, the softmax function $\mathbb{R}^n \rightarrow (0,1)^n$ is defined element-wise as follows.

$$softmax(x)_i = \frac{\exp(x_i)}{\sum_{x_j \in x} \exp(x_j)}$$

However for the text generation task, an additional *temperature* parameter τ is used to smooth the predicted probability distribution, yielding the following, updated softmax function.

$$softmax(x, \tau)_i = \frac{\exp(\frac{x_i}{\tau})}{\sum_{x_j \in x} \exp(\frac{x_j}{\tau})}$$

In this definition, $\tau > 0$ is a choosable hyperparameter. For $\tau > 1.0$, the predicted probability distribution is smoothed, spreading probability mass from the most likely token to less likely tokens. Conversely, for $\tau < 1.0$, the distribution is sharpened, increasing the probability of the most likely token while reducing the probabilities of less likely tokens.

Following the softmax operation with temperature smoothing, *top-k* sampling and *top-p* sampling (also known as *nucleus sampling* [140]) are often incorporated into text generation pipelines to improve the quality of sampled text. In these approaches, two hyperparameters, *top-k* and *top-p*, are used to control the selection of the next token. When *top-k* is applied, sampling is restricted to the k most probable tokens, disregarding the rest. When *top-p* is used, tokens are ranked by their probabilities, and the cumulative sum of probabilities is computed until it exceeds the threshold p. Tokens beyond this cumulative threshold are excluded from the sampling process. Both methods are designed to reduce issues such as sampling unlikely, outlier-like tokens while addressing common problems in greedy sampling, such as repetition and lack of diversity.

Within the context of the synthesis of annotated texts, the text generation of the GPT NeoX model can be controlled by these three hyperparameters and highlights a distinct trade-off, in particular for the temperature τ. In preliminary tests, generating texts with a low τ value generally leads to a consistent and correct markup text format in the text continuation, however the generated text is mostly composed of repeated sentences from the input prompt. Setting a larger τ value yields text with higher diversity and less repetitive sentences, yet it also increases the number of corrupted or broken markup encodings. These artifacts range from modest flaws like the introduction of unexpected label class names to instances that completely disobey the implicitly given markup syntax or domain-specific semantics.

4.3.4 Markup Decoding and Filtering

Since the generated text may be corrupted or express other undesired artifacts, the model output is filtered line-wise with regard to several filter conditions.

- **Missing Enclosing </s> Tags**: A simple measure to detect invalid sentences early can be implemented by verifying if a text line does not start with the <s> opening tag or end with the </s> closing tag.
- **Deduplication of Sentences**: The generated samples are collected and only added to the set of synthetic sentences if the sentence has not already been added in previous iterations.
- **Invalid Syntax**: If the generated line cannot be decoded according to the anticipated markup syntax, the text line is discarded.
- **Invalid or No Entities**: Given a successful markup decoding, individual lines are still discarded if no entities are annotated or their corresponding label class is not part of the set of label classes predefined in the input prompt.

When the model output has been filtered accordingly, the markup-encoded lines can be decoded into their plain text representations along with their NER annotation information. However, the markup decoding is stopped at the first occurrence of an invalid markup line, since in the case of a broken markup text at a certain position, the subsequently generated text sequence is expected to drift even further away from the valid markup syntax. As a countermeasure, the text generation can be reset and restarted after a certain sequence length has been reached.

4.3.5 NER Model Training

The acquired training data is used to train an NER model using a generic, transformer-based Spacy NER pipeline with pre-trained weights from three German BERT models. From a high-level point of view, this approach can be considered a knowledge distillation approach, as the generated corpus originates from the knowledge patterns of the LLM, and is further injected into a smaller and more efficient, task- and domain-specific NER model.

The whole process is illustrated in Figure 4.5.

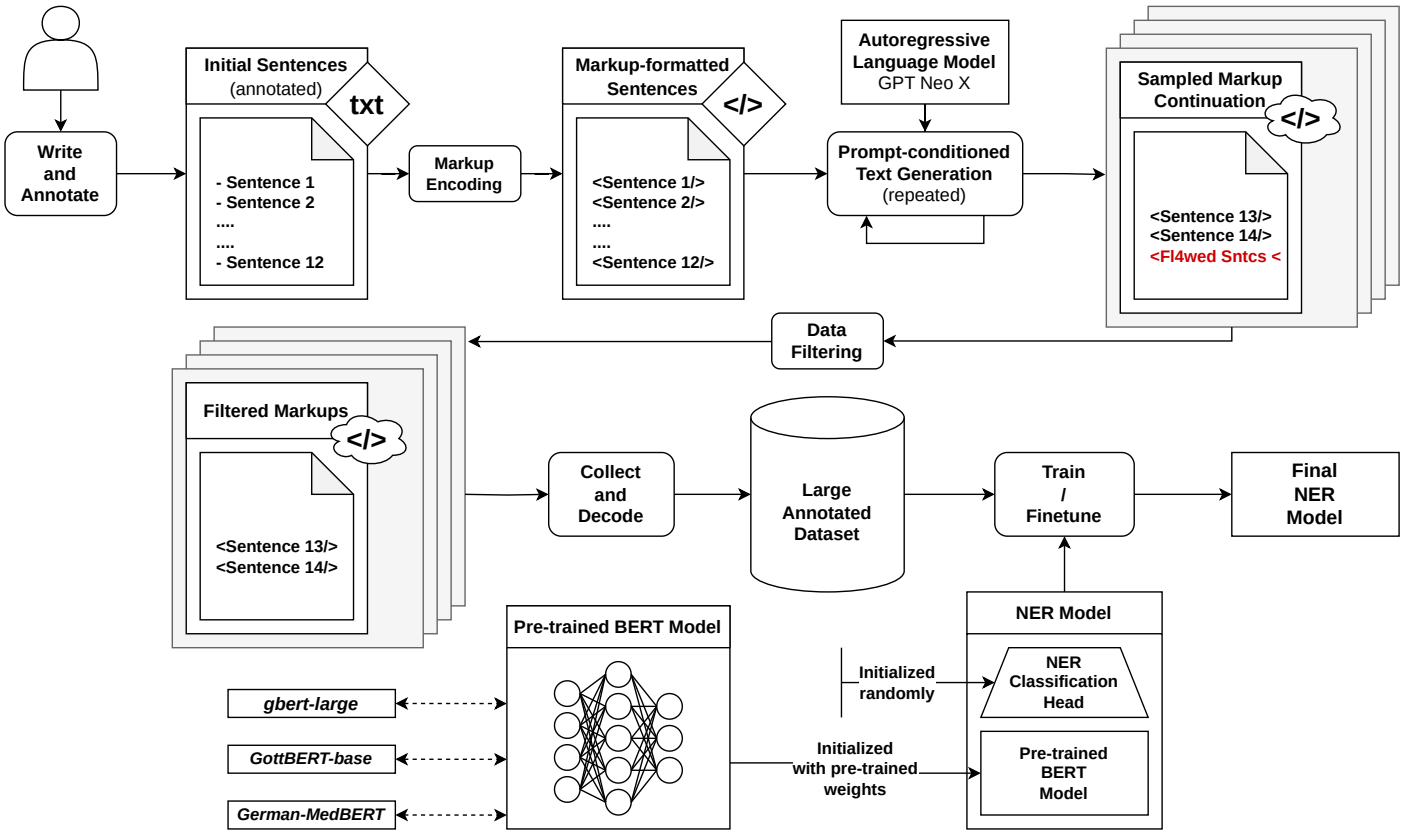

Figure 4.5: Conceptual workflow for the creation of an NER model from an annotated corpus synthesized by an LLM.

4.4 Named Entity Recognition on Weakly-annotated, WikiData-defined Corpus

Harvesting text corpora from Wikipedia is a well-established process in NLP. Due to the high quality of Wikipedia articles in terms of certain aspects like grammatical and syntactical correctness compared to other public text sources like crawled social media data, it poses a valuable resource for certain tasks such as for the pre-training of language models.

As described in previous sections of this work, two aspects are added for the use of the Wikipedia corpora.

- **Use of Annotation Information**: The actual page information of Wikipedia articles is encoded in the WikiText format, providing additional context to the plain text, such as separated content like table-oriented info boxes, or explicit references to other Wikipedia pages for particular mentions that are rendered as hyperlinks in the browser. These references can also be regarded as span-wise annotations and subsequently added to the extracted corpus as such.
- **Aligning WikiData with Wikipedia**: While Wikipedia resembles a rather unstructured, page-oriented data source, WikiData poses a highly structured, machine-readable, and graph-oriented knowledge base. Aligning individual WikiData entities with their corresponding Wikipedia pages facilitates the use of graph-based WikiData operations, such as SPARQL queries, on the Wikipedia page structure. This alignment can be particularly useful for selecting a relevant subset of Wikipedia pages that correspond to a subset of WikiData entities obtained through a custom SPARQL query.

The annotated corpus extraction process is illustrated in Figure 4.1.

To motivate the current approach, it is worth mentioning that, in the context of NER, several issues are reasons for concerns about the dataset acquisition process.

- **Existence of Raw Text Data**: For obvious reasons, any annotated corpus requires one or multiple text samples. However, in certain domains such as medical applications, obtaining text e.g. from EHR may be severely restricted for privacy reasons.
- **Existence of NER Annotations**: Assuming that raw text is available, additional NER annotations on the raw text are essential as ground truth label data for supervised learning. If no NER annotation is available, manual annotation efforts are cumbersome and costly.
- ***Matching* Text Style and Language**: In ideal scenarios, the language of the underlying text matches the language of the target application. Yet if this requirement cannot be achieved, it poses challenges regarding cross-lingual transfer. In similar terms, shifts and biases between a training corpus and the anticipated text in the

desired target domain can also negatively influence the final performance of a trained model.

- ***Matching* NER Annotations**: Depending on the specific target task that addresses a particular use case, a corpus with the pre-existing NER annotations may not match the entity types that are of interest for the use case. Furthermore, even if the annotated entities seem to match the potential entities, subtle divergences between the annotation guidelines and the expected annotation style for the particular use case can pose challenges.

The research objective of this section addresses the question of how well Wikipedia-based, annotated corpora can be automatically synthesized in combination with WikiData to be used as a training set for a transformer-based NER model for NER tasks, in particular for medical NER. Similar to previous sections, the target use case of medication detection is investigated, which allows a comparison to other approaches and can be evaluated using existing, external datasets that contain medication-related annotation information.

The proposed approach has been published as scientific work [141]. Due to specific choices of experimental parameters, the resulting medication detection model is referred to as the *ATC* model.

4.4.1 Adjustments for Obtaining a Weakly-annotated Corpus

In contrast to previous Wikipedia-based corpus extraction mechanisms, the data parsing pipeline is updated to use the WikiText parser, *wtf_wikipedia*, in combination with a refined text extraction algorithm that dynamically fits the nested block structure through a recursive block processing approach. In addition, single sentences from the WikiText-encoded text blocks are not merged and re-split into individual sentences again, but the provided sentence splits are directly adopted by the processing pipeline.

Regarding the annotation extraction, the annotation data is clustered into individual label classes because the exact entity identifiers are only needed for EL tasks and are not relevant for NER. To define a custom label class, a label class title along with a valid SPARQL query is required. Whether an actual, annotated mention in the text belongs to a label class depends on the fact whether its entity identifier is part of the SPARQL result set.

4.4.2 Annotation Imputation using Self-Training Mechanism

As previously mentioned, raw annotation data from an extracted Wikipedia corpus consists of only weakly-annotated data. However, naïvely training directly on such a dataset may lead to an imbalanced and unfavorable NER model with rather low recall scores due to the fact that not every mention is annotated strictly. For instance, in Wikipedia articles, concepts are usually annotated at their first mention, and are less likely to be annotated in later mentions within subsequent sentences. However, concepts are not guaranteed to be annotated at their first mention at all.

The first mitigation towards reducing missing annotation information is realized by the data acquisition process. Due to the sentence splitting applied by the WikiText parser, individual sentences are only extracted if a sentence contains at least one annotated entity which is part of a SPARQL result set, and hence, part of a custom label class. This measure addresses the problem of latterly unannotated mentions with a formerly annotated mention of a certain concept in several cases, yet, it cannot fully prevent it in certain edge cases. For instance, an unannotated mention of a certain concept may occur in a sentence that also contains an annotated mention of another concept. During sentence extraction for a SPARQL-defined label class, this situation can lead to the inclusion of such text samples with partly missing annotation information in the corpus, especially if both concepts are intended to be part of the label class. Consequently, the true token-wise label class, or the true lack of it, remains unclear in unannotated sections of a sentence.

A key element of the data acquisition approach concerns the fact that during the extraction of sentences of relevant entities also the spans of other annotated entities, which are not supposed to be part of any SPARQL-defined label class, are preserved. Consequently, any given token position within the dataset can be stratified into one of three distinct states with regard to the annotation uncertainty in the text sequence from the extracted corpus.

- **Positive Token Position**: The token is part of an annotation that belongs to a SPARQL-defined label class.
- **Negative Token Position**: The token is part of an annotation that does **not** belong to any SPARQL-defined label class.
- **Unknown Token Position**: The token has no annotation assigned, and thus it appears to not belong to any label class. However, due to the weak annotation, it is unknown whether the token may indeed belong to one of the relevant label classes.

This information can be utilized to improve the learning procedure by two joint techniques. First, the explicit availability of both positive and negative token annotation information facilitates the use of *contrastive learning*, by which a potential NER model is trained to differentiate between positive token and negative token states. As in both cases, the certainty about the annotation is high, and thus, the training signal at these token positions can be considered reliable, the technique contributes to a stabilized training process. Second, an *adaptive loss scaling* is introduced to adequately address both potential imbalances between positive and negative token positions as well as the handling of unknown token positions. To this end, the token-wise loss is multiplied by a certain scalar value ω to modulate the effect of the gradient update at individual token positions. Therefore, the respective parameters ω_{pos}, ω_{neg} and ω_{unk} designate the loss scaling factors, and are set to the following values

$$\omega_{pos} = \frac{\#tokens_{neg}}{\#tokens_{pos}} \quad \text{and} \quad \omega_{neg} = \frac{\#tokens_{pos}}{\#tokens_{neg}}$$

where $\#tokens_{pos}$ and $\#tokens_{neg}$ denote the number of the positive and negative tokens in the corpus respectively. This measure balances the loss at token positions in which the actual annotation is *known*. Since within the context of this work, only a single label class is used, no further adjustments towards individual label class weightings are discussed here for ω_{pos}. However, regarding the parameter ω_{unk}, a plausible choice of a right loss scaling value is not an obvious decision. Henceforth, this parameter is treated as a freely choosable hyperparameter.

The setup suffices for the training of a customized NER model on the weakly-annotated corpus. As the contrastive, namely positive and negative, token positions are balanced by the loss scaling, the remaining ω_{unk} parameter can be set to a lower value to adequately model the uncertainty about the actual annotation state in the unknown token positions. Therefore, the training process draws most information only from the known token positions.

Given that an NER model is successfully trained in subsequent steps, that model can be in return re-applied to the weakly-annotated corpus to detect the initial label classes, and thus it can transform the weak annotation into a synthetically-annotated, yet fully-annotated corpus. Because the obtained NER model only serves for the purpose of the annotation imputation step, the use of the training corpus for both the model training and its re-use during the annotation imputation does not raise typical methodological concerns regarding the use of training samples beyond the initial model training.

The overall process is illustrated in Figure 4.6.

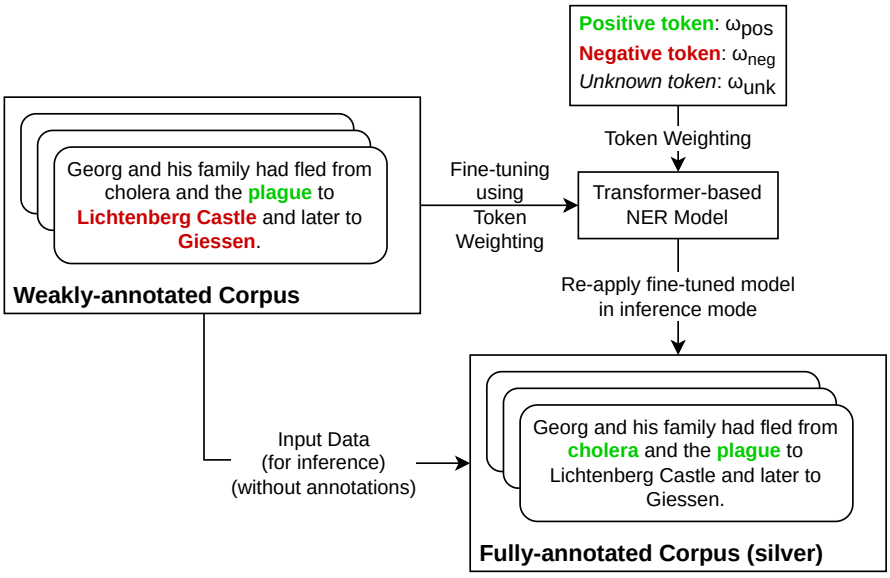

Figure 4.6: Illustration of the annotation imputation for weakly-annotated Wikipedia corpora. The red parts in the example texts illustrate negative tokens. The green parts designate the tokens that are associated with an SPARQL-defined label class.

4.4.3 NER Model Training

To evaluate the quality of the obtained corpus and thus the effectiveness of the overall approach, a generic transformer-based NER model is trained on the corpus with silver-standard annotation data. A classical NER training process without any loss scaling is applied as the annotation-imputed training corpus is assumed to be fully-annotated already, effectively rendering the current step akin to a usual NER training procedure on a manually annotated corpus.

Chapter 5

Results

The presented approaches are evaluated in this section. Due to the historical order of the development of individual works, certain evaluation materials were not available or accessible to be included in the evaluation process. Several results presented in this chapter directly resemble the measured scores calculated for their corresponding publications at their particular time. Therefore, only an inexact reconstruction of the obtained results may be feasible for experiments that take external sources into account. For example, changes in the WikiData structure may happen on a frequent basis, and thus, WikiData yields varying SPARQL responses at different points in time. Since the presented results are tightly coupled with their corresponding publications, different evaluation strategies are used to serve individual requests from peer reviews. To address this issue, all proposed methods are evaluated comprehensively in the latter part of this section using a unified evaluation pipeline on several datasets.

In this work, the overall evaluation concerns the detection of German, medical entities in unstructured documents. In particular, the commonality of all introduced approaches lies in their conceptual ability for drug detection within an NER-related setting. Hence, the drug detection task provides the common ground for all approaches.

5.1 Entity Linking via Knowledge Base-grounded Search

The search-based and knowledge base-driven approach for detecting medical entities serves as a baseline method. While the implementation relies on OpenTapioca and is primarily built for EL tasks, the method can be simplified into an NER setup by converting the spans of linked entities into label class spans for each linked knowledge base item which is considered part of the label class.

5.1.1 Reference Annotators

To compare the results in the NER setting, two alternative annotators are employed to quantify the performance of the developed approach to existing tools.

- **Apache cTAKES**: Apache cTAKES [52] is an open-source software for NLP tasks, optimized for processing medical texts. It uses the Apache UIMA structure

to represent the enriched text along with additional information generated by individual text processing components that are part of the cTAKES application stack, and can utilize ML-based as well as rule-based techniques. The primary focus of cTAKES is its application to English texts. While in theory, individual components could be improved to also support German texts, the main efforts remain directed towards the support of English text in practice. In order to run cTAKES productively, access to the UMLS metathesaurus must be gained by the user, since cTAKES heavily relies on the UMLS data as a knowledge source. Because UMLS consists not only of English terminology but also includes German terminology, with German being the second most frequent language in UMLS, cTAKES can still be expected to work on German text to some extent as a vocabulary-backed tagging approach, especially with regard to the German translation of MeSH terms being part of the UMLS database. For medication detection, cTAKES assigns an annotation layer with *MedicationMention* items to the processed documents. These items are therefore extracted and further considered as medication-related spans throughout the comparative evaluation setup.

- **PubTator**: PubTator [53] is a web-based tool for extracting biomedical concepts from texts by the National Center for Biotechnology Information (NCBI). The concepts are grouped into the supported categories `genes`, `diseases`, `species`, `chemicals` and `mutations`. The `chemical` label class is detected by a vocabulary-based lookup approach using MeSH terms and is considered semantically most relevant to a medication detection task. Therefore, it is used as such in the comparative evaluation setup.

5.1.2 Configuration for Linguistic Filtering

To limit the number of erroneously detected entities, and thus, reduce the false positive rate, the initially detected entities are filtered according to a specific set of rules.

- **Stopwords**: At least one token from the detected span should not be a stopword.
- **Part-of-speech**: At least one token from the detected span should be a noun or a proper noun.

5.1.3 Datasets

The evaluation corpora are composed of the GSC Medline and EMEA sub-corpora. While these corpora are rather focused on biomedical elements, a modified version of the first *GERNERMED* dataset iteration, acquired as described in Section 4.2, represents label classes closer to clinical text, and thus, is also included in a heavily modified way. In this case, 50 samples are randomly selected from the initial dataset, and manually checked for correctness with regard to the medication information spans.

5.1.4 Limiting Entity Sets

The EL system is set up to annotate entities beyond only medication and drug names. Since the evaluation is performed on a span-wise NER benchmark, it may be necessary to restrict the obtained set of annotated entities into a subset of entities that are considered medication-related entities, depending on the annotation schema of the evaluation dataset. Therefore, an additional post-processing step is applied, which employs a WikiData SPARQL query to determine the set of relevant entities.

Depending on the dataset, different filtering adjustments are implemented. Concerning the GERNERMED-based corpus, only found entities are passed to the evaluation script as annotated spans if a found entity is part of the following SPARQL query.

```
SELECT ?item WHERE
{
    {?item wdt:P279+ wd:Q12140 .}
    UNION
    {?item wdt:P31+ wd:Q12140 .}
    UNION
    {?item wdt:P267 ?atccode .}
}
```

At the time of evaluation, the applied filter captured medication (Q12140) instances along the edges *instance-of* (P31) and *subclass-of* (P279) and hence, encompassed an approximate set of medication entities similar to medications from GERNERMED's `Drug` label class. Due to substantial changes in the WikiData structure, the query is updated to also include entities with an assigned ATC code to improve the coverage of medication items, and thus, is not identical to the query used in the initial evaluations on GSC Medline and EMEA.

Because the performance of the EL system heavily relies on the data quality of the WikiData graph, lack of certain information may impede the observed performance during evaluation. To better estimate how the method can perform in case of sufficiently available information, the datasets are modified for a *filtered* evaluation setting. Because the label class `CHEM` (chemicals) from the Medline and EMEA corpora has additional UMLS codes as grounding assigned to individual annotation spans, only spans with UMLS codes, which also have been assigned, and thus, are known to WikiData entities, are preserved in the filtered evaluation setting. In that setting, UMLS codes unknown to WikiData are disregarded, and affected annotation spans are removed from the datasets. To obtain the set of all UMLS codes (*cui*) known in WikiData, the following SPARQL query is used.

```
SELECT ?item ?cui WHERE
{
    ?item wdt:P2892 ?cui .
}
```

5.1.5 Evaluation

The experiments were conducted using cTAKES in version 4.0.0.1 and the UMLS 2021AA release. The public endpoint[1] of PubTator was queried between May 13th and 18th of May, and on June 1st, 2022. The pre-trained snapshot of the EL system is dated November 13th, 2021. The set of WikiData entities with assigned UMLS codes were determined using the official WikiData SPARQL endpoint[2] on November 15th, 2021. Likewise, the set of medication-related entities was queried on July 31, 2024.

Quantitative Analysis

First, the approach is evaluated by precision, recall and F1-scores. Concerning these measurements, the scores are determined using the annotation sequences of individual samples, framing the task as a character-wise classification task. The characters in the annotation sequences are encoded by a label class-based (IO) tagging scheme, and the sequences are concatenated into single, large sequences to avoid potential zero-division issues in case of empty annotation sequences.

The obtained results are shown in Table 5.1.

Dataset	Method	Precision	Recall	F1-score
GERNERMED	cTAKES	.858	.512	.641
	PubTator	.760	.481	.590
	Ours	.935	.624	**.749**
Medline	cTAKES	.806	.307	.444
	PubTator	.449	.420	**.434**
	Ours	.693	.139	.232
EMEA	cTAKES	.834	.357	**.500**
	PubTator	.522	.211	.301
	Ours	.833	.172	.285
Medline (filtered)	Ours	.634	.444	***.522***
EMEA (filtered)	Ours	.604	.636	***.620***

Table 5.1: Results of the EL system for NER tasks on different datasets. PubTator and cTAKES are shown for comparisons. The scores were computed as a character-wise label class classification task.

The obtained scores highlight certain divergences between the GSC datasets and the GERNERMED samples as a clear score difference can be observed across all methods.

[1] https://www.ncbi.nlm.nih.gov/research/pubtator-api/annotations/annotate/submit/All (HTTP POST endpoint, accessed September 5th, 2024)

[2] https://query.wikidata.org/sparql (HTTP POST endpoint, accessed September 5th, 2024)

For the GSC corpora, no method can exceed an F1-score of .5, with the EL system as the worst performing method. Low recall rates underscore potential annotation divergence issues. For the EL system, the score on the GSC corpora improves if only entities with UMLS codes known in WikiData are considered as annotations, which leads to a more balanced precision-recall ratio, and hints to the fact that certain WikiData entities may lack metadata of certain coding systems. Regarding the GERNERMED samples, the performance of all applied methods improves. Here, the EL system outperforms other methods.

Qualitative Analysis

Beyond the quantitative evaluation, looking deeper into the prediction of individual samples highlights certain aspects, partly complementary to the quantitative scores.

- **Missed WikiData Entities**: Certain entities and terms that occur as annotated terms in the ground truth data have not been indexed during the initialization of the EL system, and thus, are not detected during the subsequent text processing. While their corresponding entities are present in the WikiData graph, for instance "Neurobloc" (Q29006215) and "Valdoxan" (Q29006623), these entities are rather described as "pharmaceutical product" rather than a medication entity, do not have an assigned ATC code, and generally only have rather a low connectivity to other entities in the graph. In related context, due to informal terminology certain terms may be missed even though their corresponding WikiData entities are indexed. For instance, "Vanco" is not detected by the EL system although the entity "Vancomycin" (Q424027) has been indexed.
- **Biomedical Entity Mismatch**: The sample-wise investigation confirms the low recall rate of PubTator, especially for the clinical text of the GERNERMED samples. Its term detection seems to focus on biomedical-related chemicals rather than just common medication names. For instance, it detects "Kreatinin" contrary to the ground truth data. Similarly for cTAKES, certain generic terms such as "Tablet" are categorized into the *MedicationMention* group even though it describes the administration form rather than an actual medication.
- **Non-German Term Detection**: PubTator and cTAKES also tend to struggle with German terms of generic concepts like "Antibiotikum". For cTAKES, its lack of German text structure leads to the incorrect detection of German stopwords as medication mentions. (e.g. "hat", "pro", "Das") In the case of German compound words, even the EL system may struggle to recognize certain terms. ("Antibiotikakurs")
- **Ground Truth Disputes**: In some instances, ground truth spans may prompt disputes or are erroneous. For example, "Zeffix" is recognized by the EL system but is not annotated according to the ground truth data.

As a commonality, all approaches are clearly limited in particular when it comes to the detection of unseen medication names, as this is an inherent limitation of search-based

and vocabulary-bound entity detection strategies.

Individual excerpts from the datasets are provided in Figure 5.1 and Figure 5.2 (covering GSC EMEA samples), as well as in Figure 5.3 (covering the GERNERMED samples).

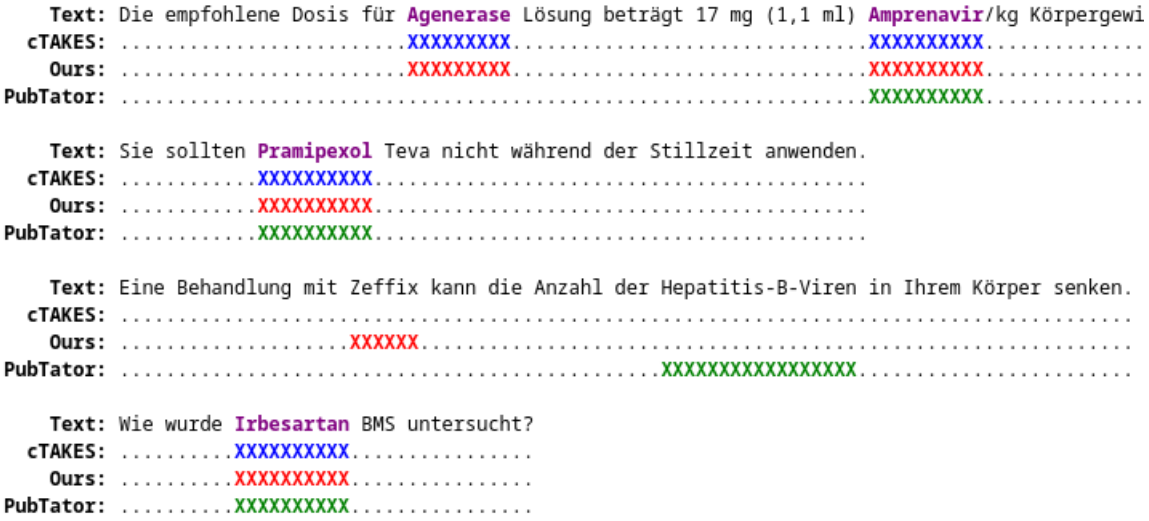

Figure 5.1: Samples from the GSC EMEA dataset with predominantly correct predictions. A colored text indicates a present annotation span from the ground truth. The colored `X` characters designate predicted annotation spans. Individual colors are tied to an annotator. The text is cut off on the right side.

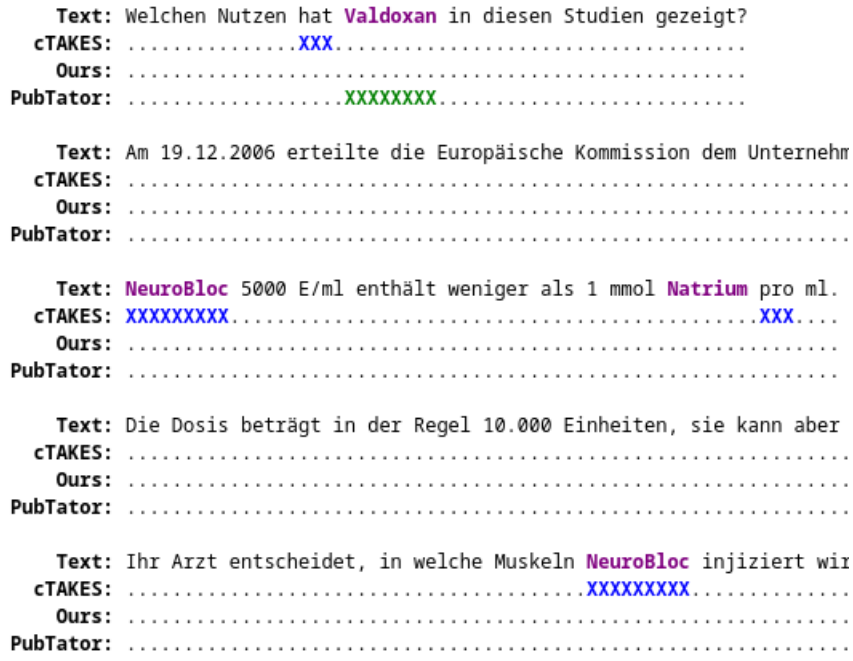

Figure 5.2: Samples from the GSC EMEA Dataset with predominantly incorrect predictions. A colored text indicates a present annotation span from the ground truth. The colored `X` characters designate predicted annotation spans. Individual colors are tied to an annotator. The text is cut off on the right side.

Figure 5.3: Samples from the (modified) GERNERMED dataset. A colored text indicates a present annotation span from the ground truth. The colored `X` characters designate predicted annotation spans. Individual colors are tied to an annotator. The text is off on the right side.

5.2 Named Entity Recognition using Translation and Annotation Projection

The n2c2 2018 ADE Track 2 corpus, serving as the source dataset for this study, is processed with only the 303 training samples for the first iteration (also referred to as *GERNERMED*), and with all 505 samples for the second iteration (also referred to as *GERNERMED++*).

5.2.1 Removal of Label Classes

Certain label class instances are removed from the annotation data. The reasons for the individual label classes are as follows:

- **ADE**: The label class concerns adverse drug events, hence their spans mark events of adverse drug reactions reported in the text and thus, is closely tied to the objective of the n2c2 ADE challenge. In contrast to other label classes with clearer semantics, ADE may require knowledge of drug use and clinical interpretation to correctly identify the relevant spans. This complexity introduces variability and potential subjectivity to the annotation process, as the identification of adverse drug events can be nuanced and heavily text-dependent. Removing the ADE label class helps to ensure that the remaining data is more reliable and uniformly interpretable, facilitating more robust and precise analysis and model training. The label class was removed both in the first and the second iteration.

- **Reason**: The label class concerns the mention of the justification for a drug administration. It can cover more fundamental underlying elements like signs, symptoms and diseases according to the annotation guide, and may be a source of ambiguity due to similar considerations stated about the ADE label class. The general usefulness depends on the nature of the dataset and the context, and in preliminary experiments the label class yielded low scores. The label class was removed both in the first and the second iteration.

- **Route**: The label class concerns the route of the administration of a drug. Investigating the actual instances of the label class in the text during a preliminary inspection lead to the decision to remove the label class in the second iteration, as the samples shows low diversity, yielding the value "PO" 5,356 out of the total 8,560 `Route` annotations. The text "IV" follows as the second most frequent term, accounting for 874 instances. Due to this homogeneity of the surface terms appearing in the text, the label class was dismissed in the second iteration to avoid potential misinterpretations of well-performing scores of the label class on the test set. Because the label class diversity in the test set mirrors its diversity in the training set, rather high-performing scores may mislead readers about the generalizability of the trained model on this label class.

5.2.2 Obtained Corpora Statistics

As described earlier, the original English text corpus is loaded along with the annotation information and preprocessed using the mask replacement. Individual sentences with at least one entity are extracted and their plain text translated into German. The translation model uses the default parameters of the *fairseq* implementation, with its beam search parameter set to five to increase the translation quality at the expense of increased computation and memory usage. The sentence pairs of English and German texts are processed further on the word alignment methods to project the English annotation information onto the German text samples. For the first iteration, the hyperparmeter for the decision boundary has to be determined which is responsible to detect samples with flawed word alignments. Ten samples with flawed word alignments were randomly selected and the decision boundary was gradually decreased until all samples were detected as flawed according to the decision boundary, resulting in the hyperparameter to be set to $\tau = 1.8$.

Certain instances of label classes are removed according to the formerly explained reasons. For the first iteration, the label classes were removed prior to the text translation. For the second iteration, all label classes were initially kept untouched and filtered out as an additional preprocessing step prior to the training of an NER model. Eventually the following corpus statistics are provided in Table 5.2.

		Synthesized Dataset	
		First Iteration (using *Fast_Align*)	**Second Iteration** (using *AwesomeAlign*)
Text	#Chars	914,826	2,101,124
	#Tokens	172,695	627,641
	#Sents	8,599	16,632
Entities	#ADE	—	*1,557*
	#Dosage	3,419	6,700
	#Drug	8,305	26,003
	#Duration	409	956
	#Form	5,242	10,546
	#Frequency	4,238	9,794
	#Reason	—	*6.244*
	#Route	4,549	*8.560*
	#Strength	4,071	10.546

Table 5.2: Dataset statistics on the two corpora from both iterations of the translation-based corpus synthesis. The tokenization is performed using the German tokenizer from Spacy. Due to prior sentence splitting, each text sample resembles a single sentence.

To better estimate the generalization performance of the model, an additional handcrafted and manually annotated dataset was created as a gold standard, out-of-

distribution (OoD) set, and is subsequently used for an additional evaluation.[3] The corpus statistics is given in Table 5.3.

Out-of-distribution Set		
Text	#Chars	5,435
	#Tokens	920
	#Sents	70
Entities	#Dosage	4
	#Drug	36
	#Duration	3
	#Form	19
	#Frequency	20
	#Strength	37

Table 5.3: Dataset statistics on the out-of-distribution (OoD) set. The tokenization and sentence splitting is performed using the German tokenizer from Spacy.

5.2.3 NER Model Training

To finally estimate how well the proposed approach, that utilizes a translation and annotation projection to synthesize an annotated German dataset from non-German corpora, can perform, NER models are trained on the obtained datasets.

For the first iteration, an NER model is trained which implements the slim, Token2Vec-based architecture that is optimized for CPU-based training due to its low computational complexity. The model architecture was chosen because no access to GPU resources was feasible at the time of development regarding the first iteration. For the training, the dataset was split into three sets, namely training (80%, 6,879 sentences), validation (10%, 860 sentences) and test (10% 860 sentences). For training, the default parameter from Spacy (Version 3.0.6) were used, including the Adam optimizer with weight decay (learning rate = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$). In total, the training of the model took 10 minutes on an Intel i7-8665U CPU. The model that performed best on the validation set was taken as final model after a training session of 20,000 total training steps.

For the second iteration, three different NER models were trained using Spacy (Version 3.2.3). The first model retains the slim, Token2Vec-based NER model similar to the model architecture used in the first iteration. The two additional NER models are transformer-based Spacy models that utilize transfer-learning by relying on pre-trained transformer models as feature encoders. For these models, the German pre-trained models *GottBERT* and *German BERT* were used. Although these pre-trained encoders could be used purely as fixed feature-encoders, their weights are updated during training to allow the gradient flow to also optimize the transformer-based encoder for

[3]The dataset has been created and annotated by a physician according to the annotation guidelines of the n2c2 2018 ADE dataset, except for the label classes `ADE`, `Reason` and `Route`. The dataset is publicly available at the following URL: https://github.com/frankkramer-lab/GERNERMED-pp/blob/main/OoD-dataset_GoldStandard.jsonl (accessed September 5th, 2024)

the NER task. During training of the transformer-based models, the transformer-specific default training parameters from Spacy were employed (learning rate = 5×10^{-5}, with 250 warmup steps), along with the Adam optimizer with unchanged parametrization. The training of the models took 8–47 minutes on a single NVIDIA Titan RTX GPU, depending on the used encoder model. (German BERT: 47m, GottBERT: 26m, Slim: 8m)

5.2.4 Evaluation

The performance of the trained models is evaluated using multiple strategies which were also governed by the availability of certain datasets and annotation sequences.

Quantitative Analysis

For the first iteration, the trained model is evaluated on the test set using an exact entity-wise evaluation using precision, recall and F1-score. Both the model and the GGPOnc 2.0 baseline tagger are subsequently tested in addition on the OoD dataset using a character-wise, IO scheme-based evaluation setup. Only the label class `Chemical_Drug`, supported by the GGPOnc tagger, is used as label class related to `Drug` due to their semantic similarities, and yields an F1-score of .5370. The annotation sequence initially predicted by the model is encoded on a token level. Hence, for a token- or char-wise evaluation, the ground truth spans that do not align onto the token boundaries, are therefore contracted until the closest token boundaries are reached, or are expanded to the closest token boundaries if the former contraction strategy results in an empty span.

The main results for the model from the first iteration are provided in Table 5.4.

NER Results		Test Set			OoD Set		
		Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Label Class	Dosage	.8783	.8757	.8770	.0169	.0455	.0247
	Drug	.6733	.6617	.6674	.5521	.5377	.5448
	Duration	.6786	.5278	.5937	.0000	.0000	.0000
	Form	.9194	.8924	.9057	.7353	.1351	.2283
	Frequency	.7914	.7692	.7801	.4846	.4783	.4814
	Route	.8993	.9014	.9004	—	—	—
	Strength	.9234	.9099	.9166	.7116	.6456	.6770
	Total	.8231	.8079	.8154	.5752	.4571	.4818

Table 5.4: NER results on test set and out-of-distribution (OoD) set from the first iteration. The test set results are obtained using entity-wise evaluation with strict span matching. The OoD set results are obtained using a char-wise, IO scheme-based evaluation. The total score is aggregated by the frequency-weighted average.

The second iteration is evaluated on the test set using a token-wise, IO scheme-based evaluation setup. In addition, the OoD set is also utilized within a token-wise IO scheme-based evaluation setup. As the underlying corpus from the second iteration

differs from the corpus from the first iteration, the test sets are not identical. For comparison, also the model from the first iteration was evaluated on this test set, however due to potential shufflings of the text samples certain samples from the current test set may have been used as training samples during the training of the model from the first iteration.

The main results for the models from the second iteration are provided in Table 5.5.

NER Results		Test Set			OoD Set		
		Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
GottBERT-based							
Label Class	Dosage	.9673	.9712	**.9692**	.1250	.2500	**.1667**
	Drug	.9689	.9250	.9464	.8913	.9318	**.9111**
	Duration	.8056	.8246	**.8150**	1.000	.4000	.5714
	Form	.9481	.9699	**.9589**	1.000	.7368	**.8485**
	Frequency	.8794	.9530	**.9147**	.9231	.6154	**.7385**
	Strength	.9708	.9636	.9672	.8661	.9604	**.9101**
	Total	.9419	.9511	**.9460**	.8830	.8443	**.8515**
German BERT-based							
Label Class	Dosage	.9625	.9707	.9666	.0769	.2500	.1176
	Drug	.9688	.9329	**.9505**	.8302	1.000	.9072
	Duration	.7909	.8246	.8074	1.000	.8000	**.8889**
	Form	.9555	.9623	**.9589**	.9091	.5263	.6667
	Frequency	.8591	.9235	.8901	.4561	.6667	.5417
	Strength	.9775	.9729	**.9752**	.9535	.8119	.8770
	Total	.9382	.9466	.9421	.8170	.7877	.7878
Spacy Token2Vec-based (Slim)							
Label Class	Dosage	.9578	.9707	.9642	.1111	.2500	.1538
	Drug	.9295	.8839	.9061	.6905	.6591	.6744
	Duration	.8229	.7488	.7841	.0000	.0000	.0000
	Form	.9649	.9501	.9574	1.000	.3158	.4800
	Frequency	.8550	.9665	.9074	.4865	.4615	.4737
	Strength	.9651	.9673	.9662	.9512	.7723	.8525
	Total	.9262	.9405	.9323	.7777	.6226	.6792
Reference / First Iteration: GERNERMED (slim, w/o `Route`) (possibly tainted on test set)							
Label Class	Dosage	.9155	.9587	.9366	.0455	.2500	.0769
	Drug	.6442	.6339	.6390	.4600	.5227	.4894
	Duration	.6125	.6967	.6519	.0000	.0000	.0000
	Form	.8418	.8816	.8613	.5000	.1579	.2400
	Frequency	.7387	.9009	.8118	.3902	.4103	.4000
	Strength	.9165	.9173	.9169	.8514	.6238	.7200
	Total	.7903	.8409	.8135	.6185	.5000	.5411

Table 5.5: NER results on test set and out-of-distribution (OoD) set from the second iteration. The test set and OoD set results are obtained using a token-wise, IO scheme-based evaluation. The total score is aggregated by the frequency-weighted average. The German tokenizer of Spacy was used. The best F1-scores are highlighted in **bold**.

To further evaluate the ability of the trained models to generalize on other corpora, the evaluation is expanded to other German, external, and publicly available datasets. Concerning the annotation information from these corpora, no standardized set of label classes is provided, and thus, a major obstacle remains the fact that label classes from a certain dataset may not be present in other datasets. Therefore, only drug- and medication-related label classes are further investigated in the evaluation of the models on external datasets as label classes semantically related to medications and drugs are available. However, even though semantical overlaps exist, different annotation guidelines were subject to the dataset annotations, and thus, these label classes may not be semantically identical.

The results on the external datasets are provided in Table 5.6.

	NER Results	Drug (char-wise)			Drug (token-wise)		
		Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
GSC Medline Dataset [Drug = CHEM]							
Models	GottBERT-based	.8583	.7008	.7716	.8372	.7059	.7660
Models	German BERT-based	.8853	.6380	.7416	.8750	.6863	.7692
Models	Spacy Token2Vec (slim)	.4365	.1818	.2567	.5000	.2157	.3014
Models	GERNERMED	.4771	.2066	.2884	.4138	.2353	.3000
Models	GGPOnc 2.0	.8217	.4876	.6120	.7714	.5294	.6279
GGPOnc 1.0 Dataset [Drug = Chemicals_Drugs]							
Models	GottBERT-based	.5340	.6653	.5924	.5691	.6079	.5879
Models	German BERT-based	.5235	.6458	.5783	.5352	.5846	.5588
Models	Spacy Token2Vec (slim)	.1854	.4340	.2598	.2122	.4083	.2792
Models	GERNERMED	.0893	.3030	.1380	.0778	.3049	.1239
Models	GGPOnc 2.0	.6369	.7381	.6838	.5728	.7222	.6389
Bronco150 [Drug = MEDICATION]							
Models	GottBERT-based	.6708	.7887	.7250	.7252	.7518	.7383
Models	German BERT-based	.6840	.6770	.6805	.7300	.6370	.6804
Models	Spacy Token2Vec (slim)	.3201	.5124	.3940	.3782	.4862	.4255
Models	GERNERMED	.1547	.4777	.2337	.1481	.4820	.2266
Models	GGPOnc 2.0	.5730	.4502	.5042	.3455	.4309	.3835

Table 5.6: NER results on external datasets. The results are obtained using a char-wise and token-wise IO scheme-based evaluation. The GGPOnc 2.0 baseline tagger is shown as reference. Due to diverging label class definitions, a higher F1-score does not necessarily imply a better annotation performance.

Assessing the quantitative results, the test set performance of the GERNERMED model indicates that it can fit the data well for certain label classes such as Dosage, Form, Route and Strength, but seem to struggle on more complex label classes such as Drug, Duration, and Frequency to a certain degree. Even though the evaluation on the OoD set is differently measured, it remains apparent that the performance beyond the test set drops dramatically among all label to various extents. However, it should be kept in mind that the statistical support within the OoD corpus is rather low in general, and severely low in particular for the Dosage and Duration label classes. This may also lead to more imbalanced precision and recall ratios, that are less apparent in the test

set scores in general.

Concerning the GERNERMED++ models from the second iteration, similar effects can be observed for the Token2Vec model being structurally identical to the GERNERMED model from the first iteration. However, the scores tend to reach higher F1-score values on the test set as well as the OoD set, which indicates that the changes made to the synthesization of the German corpus from the first to the second iteration lead to more consistent annotation metadata. In contrast to the Token2Vec models from Spacy, both transformer-based models excel on the test set and are partially able to generalize well on the OoD set, providing clear evidence for the effectiveness of transfer learning-based approaches using pre-trained transformer models. Comparing further the results from the GottBERT and German BERT models, GottBERT slightly outperforms the German BERT model in nearly all label classes on the test set. The gap even widens for the OoD set, rendering the GottBERT-based model the most robust model.

On the external datasets, the observed results seem to confirm the key findings from the test set and OoD set evaluations for the Drug-related label classes. However, due to the varying Drug-related label class definitions, the scores also include noise from the label class inconsistencies.

Qualitative Analysis

Dataset Inspection As the datasets are the underlying resources for the NER models, these datasets, acquired via translation and annotation projection, can be inspected directly. Common samples are depicted in Figure 5.4 that were randomly sourced from the datasets as snapshots.

The key observations apparent in the depicted samples can be summarized as follows:

- **Excessive Spans**: The label projection in the samples from the first iteration occasionally suffers from partially flawed alignments. Most notable, expansions that exceed correct, short entity spans can be observed. This effect is mitigated in the samples from the second iteration.
- **Spans and Punctuation**: Related to the former artifact, annotation spans that are expected to only border punctuation tokens wrongfully include these tokens as part of their span in the first iterations. This artifact can be easily explained by the simplistic whitespace-only-based word tokenization prior to the word alignment procedure for the first iteration. Consequently, this artifact does not occur anymore in the annotation samples from the second iteration.
- **Missing Entities**: In the first iteration, the overall entity projection appears less robust when compared to the second iteration, as in certain situations the label spans are missing. In particular, samples that begin with the mention of a medication are likely to be missing in the first iteration, but are present in the second iteration. However, these artifacts may also occur at other locations.

```
GERNERMED  : Sie ist auf Atovaquone Suspension 1500 mg PO / NG DAILY zur Prophylaxe.
GERNERMED++: Sie ist auf Atovaquone Suspension 1500 mg PO / NG DAILY zur Prophylaxe.

GERNERMED  : Wahrscheinlich müssen Sie nach Abschluss des aktuellen Kurses eine niedrigere Dosis Bactrim einnehmen; diese n
GERNERMED++: Wahrscheinlich müssen Sie nach Abschluss des aktuellen Kurses eine niedrigere Dosis Bactrim einnehmen; diese n

GERNERMED  : Metoprolol Succinate 50 mg Tablet Sustained Release 24 Std. Sig: Three (3) Tablet Sustained Release 24 Std. PO
GERNERMED++: Metoprolol Succinate 50 mg Tablet Sustained Release 24 Std. Sig: Three (3) Tablet Sustained Release 24 Std. PO

GERNERMED  : Er wurde zunächst wegen CAP behandelt, aber weil er im Gesundheitswesen arbeitet und unter Berücksichtigung de
GERNERMED++: Er wurde zunächst wegen CAP behandelt, aber weil er im Gesundheitswesen arbeitet und unter Berücksichtigung de

GERNERMED  : Prevacid 30 mg Capsule, Delayed Release (E.C.) Sig: One (1) Capsule, Delayed Release (E.C.) PO einmal täglich.
GERNERMED++: Prevacid 30 mg Capsule, Delayed Release (E.C.) Sig: One (1) Capsule, Delayed Release (E.C.) PO einmal täglich.

GERNERMED  : Ibuprofen 600 mg Tablette Sig: Eine (1) Tablette PO Q6H (alle 6 Stunden) nach Bedarf bei Schmerzen.
GERNERMED++: Ibuprofen 600 mg Tablette Sig: Eine (1) Tablette PO Q6H (alle 6 Stunden) nach Bedarf bei Schmerzen.

GERNERMED  : Nachdem pt von Milrinon-Tropf auf Dobutamin-Tropf umgestellt und die Nachlastreduzierung mit Hydralazin und Is
GERNERMED++: Nachdem pt von Milrinon-Tropf auf Dobutamin-Tropf umgestellt und die Nachlastreduzierung mit Hydralazin und Is

GERNERMED  : Sie sollten weiterhin Prednison 30mg Daily einnehmen.
GERNERMED++: Sie sollten weiterhin Prednison 30mg Daily einnehmen.

GERNERMED  : Bitte überprüfen Sie INR täglich, bis die Dosierung von Coumadin stabiler wird.
GERNERMED++: Bitte überprüfen Sie INR täglich, bis die Dosierung von Coumadin stabiler wird.

GERNERMED  : Erhielt insgesamt 1,5 mg Dilaudid pro PCA in PACU.
GERNERMED++: Erhielt insgesamt 1,5 mg Dilaudid pro PCA in PACU.

GERNERMED  : Zunächst auf Plaquenil und Imuran, sowie auf Prednison.
GERNERMED++: Zunächst auf Plaquenil und Imuran, sowie auf Prednison.

GERNERMED  : Er ging auch in Vorhofflimmern früh in der Krankenhausbehandlung, die mit Absetzen von Dobutamin aufgelöst.
GERNERMED++: Er ging auch in Vorhofflimmern früh in der Krankenhausbehandlung, die mit Absetzen von Dobutamin aufgelöst.

GERNERMED  : Lisinopril 10 mg Tablette [* * Monat / Tag (2) * *]: Eine (1) Tablette PO DAILY (täglich).
GERNERMED++: Lisinopril 10 mg Tablette [* * Monat / Tag (2) * *]: Eine (1) Tablette PO DAILY (täglich).

GERNERMED  : 5) h / o CAD: Der Patient nimmt Aspirin und Betablocker.
GERNERMED++: 5) h / o CAD: Der Patient nimmt Aspirin und Betablocker.

GERNERMED  : Nachdem ihr Kreatinin auf 1,4 gesunken war, wurde ihr heimliches Lisinopril wieder in Gang gesetzt.
GERNERMED++: Nachdem ihr Kreatinin auf 1,4 gesunken war, wurde ihr heimliches Lisinopril wieder in Gang gesetzt.

GERNERMED  : Zu diesem Zeitpunkt wurde die CPR eingestellt und er erhielt keine weiteren Vasopressoren.
GERNERMED++: Zu diesem Zeitpunkt wurde die CPR eingestellt und er erhielt keine weiteren Vasopressoren.

GERNERMED  : Er wurde mit Coumadin und Heparin und einer Sekunde Levofloxacin begonnen.
GERNERMED++: Er wurde mit Coumadin und Heparin und einer Sekunde Levofloxacin begonnen.

GERNERMED  : Bitte titrieren Sie zur besseren Steuerung der Zinssätze Dilt auf.
GERNERMED++: Bitte titrieren Sie zur besseren Steuerung der Zinssätze Dilt auf.

GERNERMED  : HydrALAzine 50 mg PO Q8H 7.
GERNERMED++: HydrALAzine 50 mg PO Q8H 7.

GERNERMED  : Wir begannen mit einem zweiwöchigen Antibiotikakurs.
GERNERMED++: Wir begannen mit einem zweiwöchigen Antibiotikakurs.

GERNERMED  : Rifabutin 150 mg Capsule Sig: Zwei (2) Kapseln PO DAILY (täglich).
GERNERMED++: Rifabutin 150 mg Capsule Sig: Zwei (2) Kapseln PO DAILY (täglich).
```

Figure 5.4: Samples from the datasets from the first (GERNERMED) and second iteration (GERNERMED++). A colored text indicates a present annotation span. Encoded label classes are `Drug` (green), `Strength` (blue), `Form` (magenta), `Frequency` (red), `Dosage` (cyan), `Duration` (brown), `Reason` (bright green), `Route` (orange), `ADE` (purple). The label classes Reason and ADE are only present in the GERNERMED++ dataset. The text is cut off on the right side.

- **Translation Failures**: The certain situations the translation into German was not successful and the initial English word is preserved. These artifacts seem to happen at words with unnatural capitalization and English-specific abbreviations.

As the same translation model is used in both iterations, both corpora are equally affected.

- **Missed Pseudonymization Mask**: In one instance, a pseudonymization mask aimed to obscure a date-related information is observed was missed by the mask replacement.
- **Absence of German Prescription Coding**: With respect to the target language, German, the synthesized datasets mostly lack the coding of prescription instructions in the format *<morning>-<noon>-<evening>*, such as *0-1-1*, *1-0-1*, or similar. While this kind of prescription coding pattern is frequently used in the OoD set, no such instances could be found in the synthesized dataset. This also raises the issue of potential annotation instruction ambiguity for those instances, as they could be considered to belong to either label class `Frequency` or `Dosage`, or could even be modeled as an added, separate label class. Therefore, for the OoD set, the adopted annotation policy ignores these instances, and thus, no label class is assigned in the ground truth annotations.

NER Inspection Looking further into the annotation outputs of the NER models, the OoD corpus is utilized for the sample-based analysis. Instances of those samples are shown in Figure 5.5, which depicts the first twelve samples from the entire text samples, annotated by different NER models.

Based on the sampled subset from the ground truth-based OoD samples and their entities predicted by the NER models, the following main findings can be identified:

- **Robustness of Transformer-based Models**: The visual analysis confirms largely the ability of both transformer-based models to detect the frequent label classes `Drug` and `Strength`, as this conforms well with the measured F1-scores.
- **Semantic Behavior**: The transformer-based models appear to perform better in cases in which a rich semantic context is needed and a detection based on syntactical features is less reliable. This can be observed for instances of the `Form` (e.g. “Tablettenform” or “Filmtabletten”) and `Drug` (e.g. “Betablockern”; A change to “Betablocker” triggered the Token2Vec-based models to detect the entity.) label classes. The Token2Vec-based models also tend to spontaneously detect rather implausible spans (e.g. “sinnvoll”, “Zucker-Werte”, “1-0-0 augmentieren.”).
- **Error Propagation**: The GERNERMED model predicts spans that wrongfully include trailing punctuation characters, or that expand to adjacent but unrelated tokens. (“10 mg empfohlen.”) This artifact may be a reminiscence from the flawed synthesized dataset from the first iteration. As the dataset serves partially as training set, these errors are eventually propagated into the model.
- **Training Data-inflicted Error**: The only `Duration` entity depicted in the Figure, “lebenslang”, is ignored by all models. Investigating this artifact closer, this can be explained by the fact that no instance of the term “lebenslang” (and “live-long”

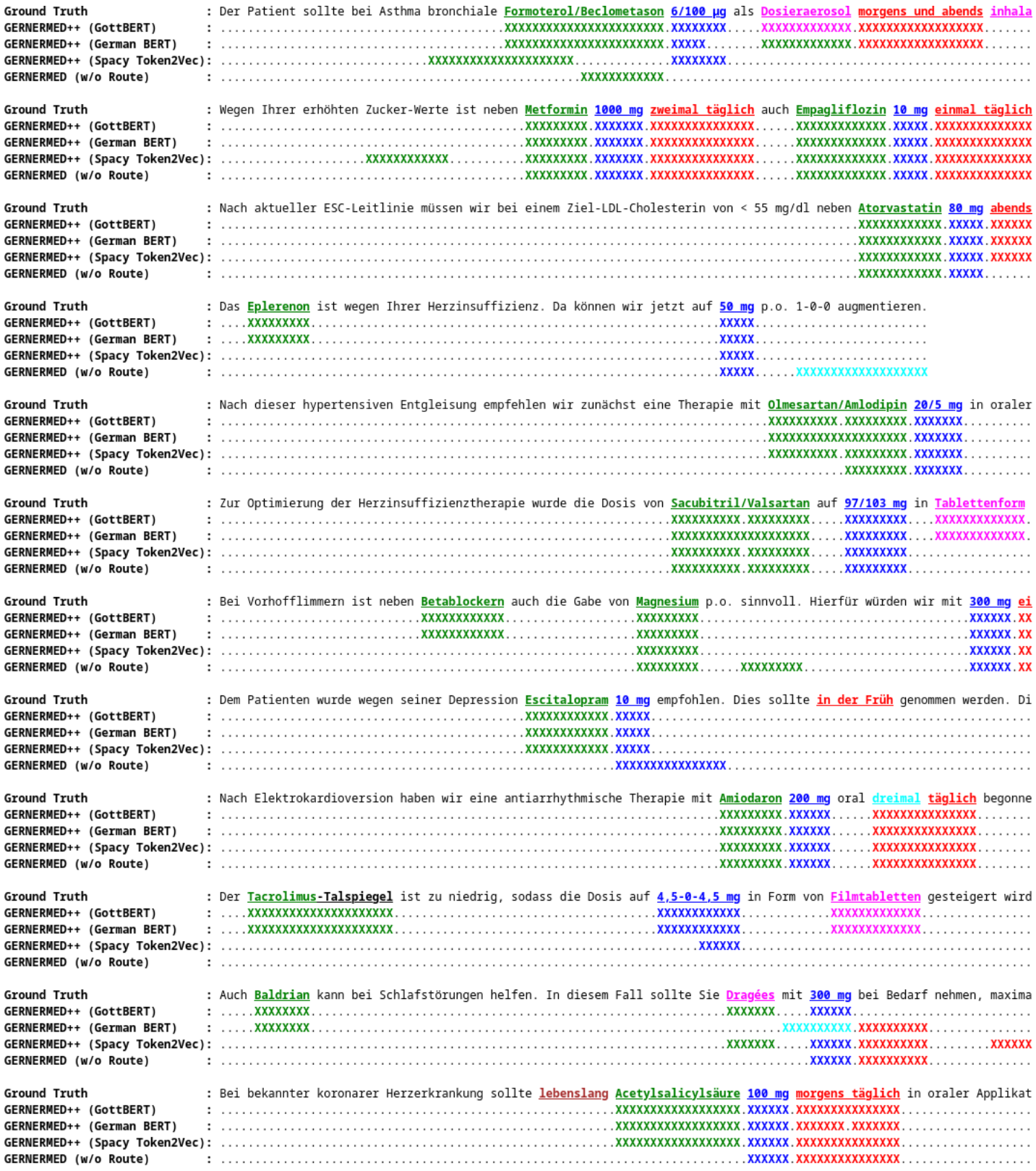

Figure 5.5: Samples from the out-of-distribution (OoD) set. A colored text indicates a present annotation span from the ground truth. The underlined text indicates the span aligned onto the less fine-grained token structure used for evaluation and training. The tokenization is performed by the German tokenizer from Spacy. The colored `X` characters designate predicted annotation spans. Encoded label classes are `Drug` (green), `Strength` (blue), `Form` (magenta), `Frequency` (red), `Dosage` (cyan), `Duration` (brown). The text is cut off on the right side.

in the English corpus) is annotated in the synthesized datasets, although the term appears in the German (and English) text samples. Consequently, this artifact is rooted in the initial English source corpus.

- **Non-obvious or Challenging Annotations**: In some instances, the correct annotation may not be immediately apparent, or may come with additional complexity. For example, "in der Früh" as a southern German expression, pose a challenge to all NER models. Other cases such as "Draggies" may be even identified as a `Drug` entity by an inexperienced person. Similarly, instances like "dreimal täglich" could be perceived as a `Frequency` entity although the correct annotation is likely to split the span into a `Dosage` ("dreimal") and a `Frequency` ("täglich") span.
- **Ground Truth Inconsistencies**: The OoD set also exhibits inaccurate annotations, and is partially inconsistent. For instance, "Bei Bedarf" is commonly annotated as a `Frequency` entity, but is not annotated in the OoD set. Other instances such as "zweimal täglich" or "einmal täglich" are only labeled as `Frequency` spans but could also be split into two separate `Dosage` and `Frequency` spans. However, the latter inconsistency can also be observed in both synthesized datasets.
- **Adjacent Spans**: Related to inconsistencies in the OoD ground truth data, adjacent medication mentions separated by a slash are annotated by a single, large span, however such instances should be encoded as multiple, separate spans. Only in limited contexts, the prediction yields two individual `Drug` entities instead of one large entity. (e.g. "Olmesartan/Amlodipin" by GottBERT, Token2Vec) Similarly, certain `Strength`-related entities suffer from these effects. However, because the predicted annotation sequence operates on token level and the tokenizer from Spacy does not always split hyphenated compound words into different tokens, certain instances cannot be encode by multiple individual spans. (e.g. "4.5-6-4.5 mg")

5.3 Named Entity Recognition using LLM-generated Annotated Corpus

As discussed in the previous chapter, the GPT NeoX 20B model is used in order to generate an annotated dataset. For this study, the parameters of the experimental setup are provided first, and the evaluation results are presented as a second step.

5.3.1 Dataset Sampling

Prompt Design

The input prompt is designed in a way to annotate the label classes `Medikation`, `Dosis` and `Diagnose`. Because the GPT NeoX language model is pre-trained on large text corpora, but is not fine-tuned further on instructional corpora, it does not naturally align to a specific prompt template, but is largely optimized for basic text continuation tasks. Therefore, the annotated text samples encode the information about the label classes by their semantic expression as well as by their actual usage within the input prompt, however no further definition on the label classes is provided.

In total, the input prompt consists of twelve text samples encoded in the proposed markup structure that allows for the embedding of entity spans in plain text. To indicate the model to continue with another text sample, a trailing opening tag `<s>` is added to the end of the input prompt. The complete input prompt is given in Figure 5.6.

```
1  <s>Zur weiteren Bekämpfung des <class="Diagnose">Juckreiz</class> wird die Einnahme von täglich <
       ↪ class="Dosis">100mg</class> <class="Medikation">Cortison</class> empfohlen.</s>
2  <s>Bei wiederkehrender Infektion wie einer <class="Diagnose">Sepsis</class> oder schweren <class="
       ↪ Diagnose">Pnseumonien</class> wird eine Überwachung erforderlich sein.</s>
3  <s><class="Medikation">Valsartan</class>/<class="Medikation">HCT</class> <class="Dosis">160</class
       ↪ >/<class="Dosis">12,5 mg</class> 1-0-0</s>
4  <s><class="Medikation">Pantoprazol</class> <class="Dosis">40 mg</class> p.o.</s>
5  <s>Die feingewebliche histopathologische Untersuchung ergab den Befund einer <class="Diagnose">
       ↪ Metastase</class> des bekannten malignen <class="Diagnose">Melanoms</class>.</s>
6  <s><class="Diagnose">Diabetes Typ 2</class>-Patienten müssen regelmäßig <class="Medikation">
       ↪ Insulin</class> (mindestens mit <class="Dosis">12ml</class> dosiert) spritzen.</s>
7  <s>Ich nehme <class="Medikation">Antibiotika</class> seit Tagen. Seitdem ist die <class="Diagnose
       ↪ ">Mandelentzündung</class> deutlich besser geworden.</s>
8  <s>Entlassung: <class="Dosis">40mg</class> <class="Medikation">Lidocain</class> wegen <class="
       ↪ Diagnose">Kopfschmerzen</class></s>
9  <s>Zusammenfassende D: Zervix-PE bei 11 und 2 Uhr mit ausgeprägter <class="Diagnose">chronisch-
       ↪ florider Zervizitis<class="Diagnose">.</s>
10 <s>Die Verschreibung von <class="Medikation">Hämatokrin</class> <class="Dosis">43mg</class> war
       ↪ unnötig.</s>
11 <s>Der Patient klagt über <class="Diagnose">Karditiden</class> und nimmt täglich <class="
       ↪ Medikation">Nifedipin</class> ein.</s>
12 <s>D: PE-Material der Portio bei 1 Uhr mit Nachweis einer schwergradigen <class="Diagnose">squamö
       ↪ sen intraepithelialen Läsion</class> (<class="Diagnose">HSIL</class>; hier noch %<class="
       ↪ Diagnose">CIN II</class>).</s>
13 <s>
```

Figure 5.6: Input prompt for the text generation task. The sentences are encoded according to the markup scheme. The trailing <s> indicates the beginning of a new sentence to the model. Only the shown sentences are used for the sentence sampling as this input prompt is directly used as input for the GPT model.

Text Sampling

The temperature parameter is set to $\tau = 0.8$ and top-p to $top_p = 0.9$ for 1,000 text generation cycles, in addition to 100 cycles with the temperature increased to $\tau = 0.9$. For every text generation cycle, a maximum of 768 tokens are sampled. The inference is performed on multiple NVIDIA A100 GPUs in parallel, and took a total of 118 h of compute time.

After inference, a total of 17,776 raw samples were extracted from the generated text using basic text parsing, with no further validation performed at this initial stage.

Dataset Decoding and Filtering Statistics

The raw data samples are further parsed and validated according to the filter conditions. The effects of the sequentially applied filtering steps are provided in Table 5.7.

Filter	#Samples	% of Baseline	Relative Impact
Baseline (raw) samples	17,776	100%	
↪ No </s> tag	16,603	93%	15%
↪ Duplicates removal	11,328	64%	66%
↪ Invalid syntax removal	11,326	64%	0%
↪ Invalid or no labels	9,845	55%	18%
Filtered samples	9,845	55%	

Table 5.7: Impact of data filtering. The majority of samples removed are due to duplicate removal. All percentage numbers are rounded.

Because a single generation cycle may produce multiple text samples that can be extracted through basic text parsing, a total of 17,776 samples make up the initial dataset. Of these, 1,173 samples are removed due to the missing closing tags, potentially caused by exceeding the maximum token limit, leading to trailing text samples without closing tags. The majority of removals, however, result from duplicate entries, accounting for 5,275 instances. An additional 1,481 samples are excluded due to either having invalid classes or lacking any assigned label class. Notably, since filters are applied in sequential order, some samples with issues such as invalid label classes may be removed earlier due to other artifacts, such as missing `</s` tags, and are therefore already included in the filtering statistics of an earlier stage.

The statistics on the final corpus are given in Table 5.8. The majority of entities are from the `Medikation` label class with roughly one occurrence per sample on average. The other label classes are used less frequently, but even the least frequent label class `Diagnose` is mentioned roughly once for every second sample on average, indicating that all predefined label classes are actively used according to the corpus statistics.

Final Dataset		
Text	#Samples	9,845
	#Chars	693,467
	#Tokens	121,027
	#Sents	10,663
Entities	#Dosis	7,547
	#Medikation	9,868
	#Diagnose	5,996

Table 5.8: Dataset statistics on the filtered final dataset. The tokenization and sentence splitting is performed using the German tokenizer from Spacy.

5.3.2 NER Training

Similar to previous sections, the generated corpus is used to train several NER models. In particular, three pre-trained transformer models, GBERT (GBERT-large), GottBERT (base) and German MedBERT, are used for the fine-tuning on the dataset. Therefore the dataset was randomly split into training (7,876 samples, 80%), validation (985 samples, 10%) and test (984 samples, 10%) sets. The transformer-based NER model from Spacy (Version 3.4.1) was used for training with the default hyperparameter configuration (learning rate = 5×10^{-5}, with 250 warmup steps) in combination with the Adam optimizer. The training took 25-55 minutes for the models to converge (GottBERT: 25m, GBERT: 55m, German MedBERT: 48m).

5.3.3 Evaluation

Similar to the strategy in the previous section, the evaluation of the trained NER models is used on one hand to estimate the usefulness of the trained models, but also serves as a proxy to estimate the quality of the generated dataset on the other hand. In contrast to an translated corpus stemming eventually from actual medical documents, a fully synthetic corpus generated by an LLM may raise further questions with regard to data quality.

Quantitative Analysis

Dataset Inspection Due to the synthetic nature of the obtained corpus, the dataset is investigated with respect to its susceptibility of potential token repetition from the input prompt without a meaningful diversification of the text.

Both the encoded text from the input prompt and the generated text is analyzed to determine the most frequent tokens while tracking their corresponding presence in the input prompt. In case that the LLM-based text augmentation cannot provide any meaningful text augmentation and only reiterates the text from the input prompt bluntly, two observations should be anticipated: First, most frequent tokens from the dataset should reappear also in the text from the input prompt. Second, the token distribution from the input prompt should follow strictly the token distribution from

the input prompt, while specific tokens, which occur frequently the input prompt but are less frequent in common texts, are similarly distributed in both the dataset and the input prompt.

The results from the text analysis addressing the most frequent tokens is shown in Figure 5.7. For the analysis, stop words are filtered using Spacy, and the text is normalized into a case-insensitive sequence. Due to occasional lemmatization gaps (e.g. "Impfungen" could not be lemmatized) in the German, table-based lemmatization component of Spacy (Version 3.7.2 with the *de-core-news-sm* pipeline in version 3.7.0), no lemmatization is applied.

Given the text analysis, the major observation indicates that several of the most frequent tokens from the synthetic corpus are actually generated by the LLM and do not occur in the input prompt. Similarly, the top-most token distribution of the synthetic corpus does not appear to match the distribution of the input prompt in an unnatural manner. Evidently, the text augmentation even introduces new domain-specific terms (e.g. medications "propranolol", "aspirin"). However, some tokens *are* in fact mentioned frequently, although they could be perceived as very specific terminology (e.g. "zervix-pe", "hsil").

Beyond the scope of the top-most tokens, the distribution of unique tokens from the input prompt and the generated text is measured to quantify the extent to which the LLM introduces new tokens that are not mentioned in the input prompt. After applying the processing steps for the token analysis, 62,520 total tokens are found to be composed of 8,794 unique tokens, and 20,842 total tokens (76 unique tokens) can also be found in the input prompt. The cumulative distribution is shown in Figure 5.8. Two aspects become apparent. First, one third of the total tokens are also found in the prompt, which could be both caused by the fact of blunt repetition but also by the legitimate use of common, and potentially domain-specific, vocabulary (e.g. "patient", "untersuchung", "mg"). Second, the analysis shows that over 99% of the unique tokens are actually introduced by the LLM and are therefore not merely drawn from the input prompt.

NER Inspection The performance of multiple NER models on the test set is evaluated using a character-wise IO scheme. Additionally, similar to the strategy from the previous section, the trained models are also evaluated on the OoD set for the `Drug` label class, which can be considered semantically related to the `Medikation` label class from the native training set, as well as the `Strength` label class using the models' `Dosis` label class, to further estimate the robustness of the results.

The results on both the test set and OoD set are shown in Table 5.9 for a character-wise IO scheme-based evaluation. Due to discrepancies in the evaluation pipeline, the scores

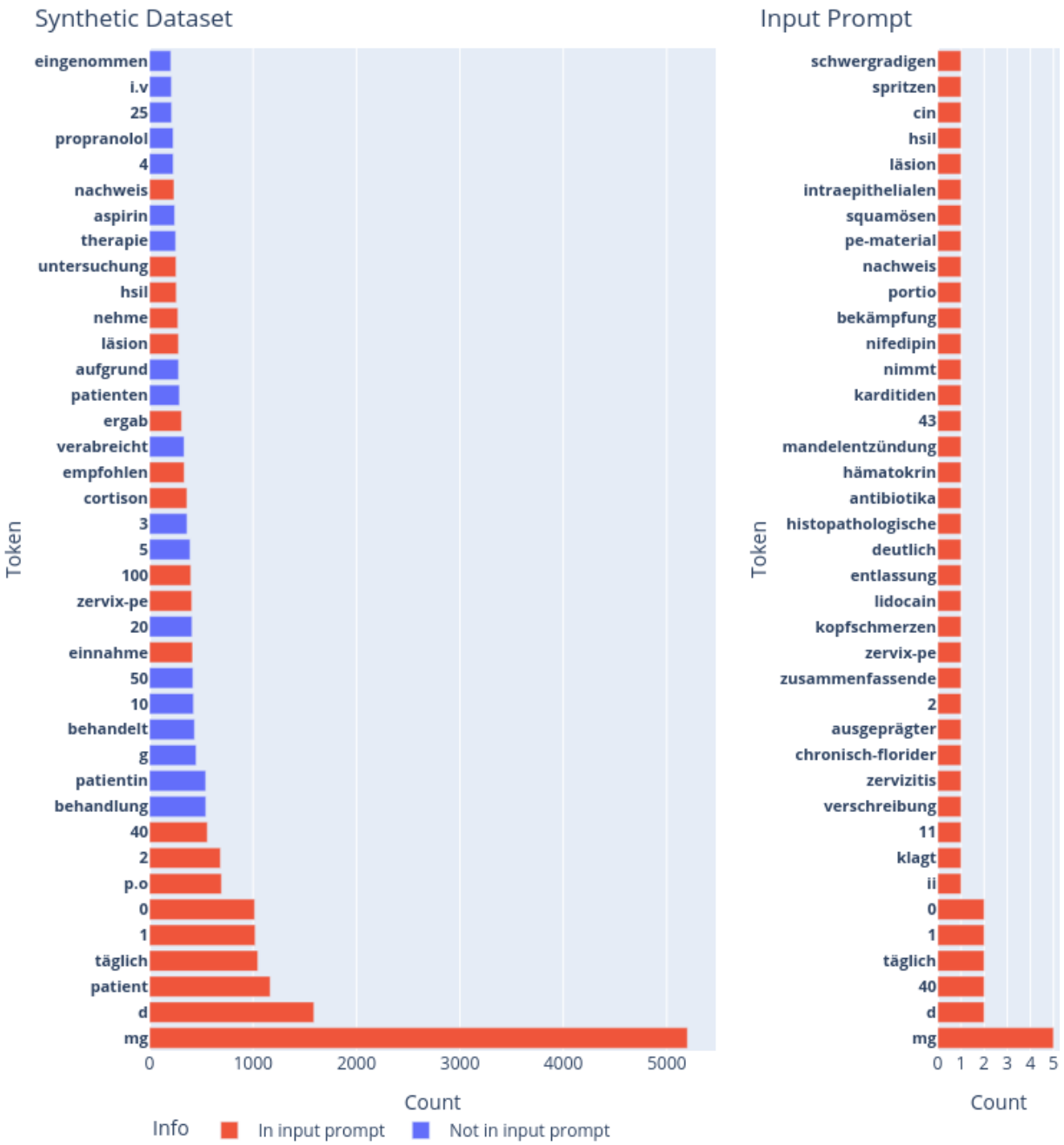

Figure 5.7: Most frequent tokens from the synthetic dataset and input prompt. Several top-most tokens are not mentioned in the input prompt. Spacy's German tokenizer is used for the tokenization. The tokens are counted case-insensitively and have been filtered for stop words and punctuation.

presented in this work differ from those reported in the reference paper [139][4].

Focusing on the medication-related label class, several external datasets are applied on the trained NER models for in char-wise and token-wise IO scheme-based evaluation runs. The results are provided in Table 5.10. The evaluated scores of the GGPOnc 2.0 NER tagger are given as an external reference model.

[4]In the context of this work, the scores have been re-computed using an updated evaluation pipeline which includes minor corrections as well as technical adjustments. However, the random train-test split could not be reconstructed due to the randomized split selection. Therefore, the scores for the test set are kept identical to the scores reported in the reference paper.

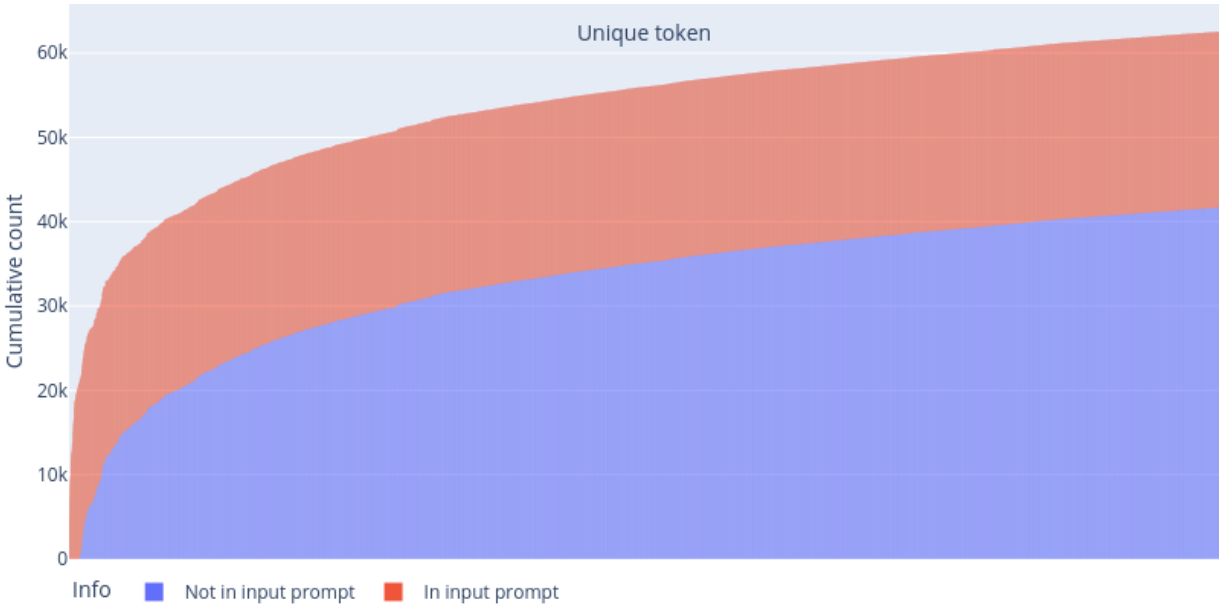

Figure 5.8: Cumulative distribution of unique tokens from the synthetic dataset. 0.9% of all unique tokens are part of the initial prompt but account for 33.3% of all tokens by frequency. Spacy's German tokenizer is used for the tokenization. The tokens are counted case-insensitively and have been filtered for stop words and punctuation.

NER Results		Test Set			OoD Set [Drug = Medikation, Strength = Dosis]		
		Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
GottBERT-based							
Label Class	Medikation	.979	.910	.943	.8392	.9481	**.8904**
	Dosis	.887	.907	.897	.7617	.9578	.8486
	Diagnose	.896	.844	.870	–	–	–
	Total	.936	.886	.910			
GBERT-based (GBERT-large)							
Label Class	Medikation	.870	.936	**.949**	.6651	.9741	.7904
	Dosis	.883	.921	**.901**	.7621	1.000	**.8650**
	Diagnose	.870	.895	**.882**	–	–	–
	Total	.918	.919	**.918**			
German MedBERT-based							
Label Class	Medikation	.980	.905	.941	.7671	.9009	.8286
	Dosis	.829	.890	.858	.6807	.9536	.7944
	Diagnose	.910	.730	.810	–	–	–
	Total	.932	.842	.883			

Table 5.9: NER results on the test set and out-of-distribution (OoD) set. The test and OoD results are obtained using a char-wise, IO scheme-based evaluation. The total score is aggregated by the frequency-weighted average. The best F1-scores are highlighted in **bold**. The test set results reported in the reference paper are shown in gray.

The obtained results on the test set indicate that the GBERT model achieves the best performance, yet the GottBERT model achieves comparable score values albeit being smaller in size and parameter numbers. The domain-specifically fine-tuned German MedBERT model however yields slightly inferior scores overall, and tends to show a slightly more imbalanced behavior with regard to the precision and recall score, in

NER Results		Drug (char-wise)			Drug (token-wise)		
		Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
GSC Medline Dataset [Drug = CHEM]							
Models	GottBERT-based	.9188	.4678	.6199	.9000	.5294	.6667
Models	GBERT-based	.7491	.7107	.7294	.7600	.7451	.7525
Models	German MedBERT-based	.7252	.4711	.5711	.7879	.5098	.6190
Models	GGPOnc 2.0	.8217	.4876	.6120	.7714	.5294	.6279
GGPOnc 1.0 Dataset [Drug = Chemicals_Drugs]							
Models	GottBERT-based	.4567	.7900	.5788	.4739	.7616	.5843
Models	GBERT-based	.3016	.8555	.4459	.3232	.8372	.4664
Models	German MedBERT-based	.5696	.6823	.6208	.5785	.6393	.6074
Models	GGPOnc 2.0	.6369	.7381	.6838	.5728	.7222	.6389
Bronco150 [Drug = MEDICATION]							
Models	GottBERT-based	.6548	.8125	.7251	.6776	.7542	.7139
Models	GBERT-based	.4618	.9108	.6129	.4650	.8636	.6045
Models	German MedBERT-based	.6172	.7054	.6584	.5746	.6617	.6151
Models	GGPOnc 2.0	.5730	.4502	.5042	.3455	.4309	.3835

Table 5.10: NER results on external datasets. The results are obtained using a char-wise and token-wise IO scheme-based evaluation. The GGPOnc 2.0 baseline tagger is shown as reference. Due to diverging label class definitions, a higher F1-score does not necessarily imply a better annotation performance.

particular for the `Diagnose` label class. On the OoD set, only the medication-related and strength-related label classes can be further evaluated. Here, the generalization ability of the German MedBERT model appears to be overall inferior to both the GottBERT and GBERT models, although it can surpass the GBERT model for the `Drug` label. On this aspect, the GottBERT model is able to surpass the GBERT and German MedBERT models on the drug detection task, and thus, flipping the lead of the GBERT model on the test set regarding F1-scores for the `Medikation` label class. Overall, all models achieve relatively high recall scores compared to their precision.

Focusing exclusively on the medication detection task, the external results show a notable decline in terms of performance scores, in particular on the GGPOnc 1.0 dataset, which is a natural phenomenon to a certain degree due to the multifactorial shifts in dataset style, annotation guidelines, and annotation quality. Yet, comparing the external results to the external results from the previous section (Table 5.6), the GottBERT model achieves only marginally inferior F1-scores on the Bronco150 corpus compared to the GottBERT model trained on the translated n2c2 corpus in the former section, but on the GSC Medline corpus the performance drops significantly for the GottBERT model. Contrary to the GottBERT model, the larger GBERT model achieves the best performance on the GSC Medline corpus and remains only moderately inferior to the GottBERT model from the previous section. However, the GBERT model does not perform well on both the GGPOnc 1.0 and Bronco150 corpora. The German MedBERT model remains consistently in the .55 and .65 value range with respect to the F1-score across all three corpora.

Beyond the scope of the F1-scores, in certain instances the precision and recall scores

tend to be severely imbalanced. On the GSC Medline corpus, the GottBERT and German MedBERT models are skews towards high precision with low recall scores, whereas the GBERT model shows a rather balanced behavior pattern. However, on the other datasets imbalances are clearly observable in a reversed balance pattern with a higher recall and lower precision score. When focusing on the GBERT and GottBERT models in Bronco150 and Medline, it appears that both models converged to different implicit label definitions and degree of definition strictness during fine-tuning on the identical training corpus.

Qualitative Analysis

Moving on to the qualitative analysis of individual samples, both the synthesized samples from the datasets and annotated NER samples from the OoD corpus are investigated.

Dataset Inspection A limited set of samples is randomly extracted from the synthesized dataset and displayed in Figure 5.9.

Several observations can be drawn from the shown dataset subset. First, the following major aspects regarding the text style are worth mentioning.

- **Repetitive Text Patterns**: Certain structural text phrases from the input prompt are reiterated in the dataset. A most notable example refers to cases of text samples starting with the ”D: “ prefix that occurs seven times in the visualized subset. However, it appears that no outright repetition of entire text blocks from the input prompt is observed.
- **Partially Semantic Blunders**: In some instances, samples are merely meaningless or incoherent word compositions, yet they adhere to the medical vocabulary. Nevertheless, basic patterns of semantic relationships are mostly preserved (e.g. “Schlaflosigkeit” $\leftrightarrow$ “Amitriptylin”, “Karditis” $\leftrightarrow$ “EKG”).
- **Broken Grammar or Syntax**: In a few instances, the sentence structure is even syntactically broken (e.g. ”wobei der Patient eine vermehrte Atemnot 50mg Nifedipin 90mg pro Tag“). Still, instances that consist purely of medication and strength/dosage information are expected, because such an instance is also part of the initial input prompt.

Second, regarding the generated annotation structure, the following artifacts are observable.

- **Inconsistent Annotation Behavior**: Instances like dosage and frequency coding phrases (e.g. “1-0-0”) are partially annotated as `Dosis` and partially left unannotated. Because such a phrase could also be found as an unannotated text within the input prompt, the generated annotations do not consistently adhere to the implicit annotation rules.

GPTNERMED: Ich würde **Tadalafil** **20mg** empfehlen.

GPTNERMED: D: **Kolonkarzinom** im Bereich der **Ileum** mit **16mg** **FOLFOX4**.

GPTNERMED: Wie stellte der **Penicillintabletten** **80mg** **4-1-1** Einnahme ein.

GPTNERMED: Krankenhausrechnung: **Enalapril** **5mg** 1-0-0

GPTNERMED: D: **Propranolol** **160mg** p.o.

GPTNERMED: Ich habe **Morphium** verabreicht.

GPTNERMED: D: Zervix-PE bei 0 Uhr. Die **Antibiotika** werden seit **3** **Tagen** verabreicht.

GPTNERMED: D: **Krampf** der **Uterus** mit **Zervix**.

GPTNERMED: Die Einnahme von **Digoxin** **0,125mg** **1,5mg** wird kontraindiziert.

GPTNERMED: Falls Sie auch **Tonsillitis** haben, sollten Sie eine **Ampicillin** oder **Amoxicillin** einnehmen.

GPTNERMED: Die Behandlung von **Hypospadie** kann unterstützt werden.

GPTNERMED: Der Patient befindet sich bereits unter Antibiotikatherapie und bekommt täglich **Ciprofloxacin** **500mg**.

GPTNERMED: **Diuretika/HCT** **160/12,5** **mg** 1-0-0

GPTNERMED: Sie fühlt sich sehr schlecht und bekommt eine Anfangsverdachtsdiagnose **Tuberkulose**.

GPTNERMED: D: Wenn die Dauer des Mosapride über 24h ist, müssen die **2x** täglich erhöht werden.

GPTNERMED: Eine Durchführung einer **Nierenbiopsie** war nicht erforderlich.

GPTNERMED: Die Patienten werden regelmäßig **Insulin** (mindestens mit **12ml** dosiert) spritzen.

GPTNERMED: Die Behandlung **Tetracycline** wurde vor 5 Tagen beendet.

GPTNERMED: Das MRT zeigt eine **Komplexe** **Läsion** im **1**.

GPTNERMED: Ich habe den Eindruck, dass die **Nifurat** **30mg** weniger wirksam war, als die **Naftidrofuryl**.

GPTNERMED: D: Zervix-PE bei 11 Uhr mit **Zervix-PE** und **Karditis**.

GPTNERMED: Der Patient klagt über **Rheumatisches** **Fieber** und nimmt täglich **Celecoxib** ein.

GPTNERMED: Bei der weiteren Verbesserung der **Schlaflosigkeit** mit **Amitriptylin** **1,25mg** 2-0-0 wird die Dosis auf **2,5mg** **3mg** erhöht.

GPTNERMED: Die Patientin wird unter Ausgang **10mg** **Aspirin** empfohlen.

GPTNERMED: Die Blutdruckwerte liegen unter **90/50** **mmHg**

GPTNERMED: Die Diagnose **Karditis** wurde bei einem EKG abgeleitet. Die Kardiologie ersucht um **40mg** **Diclofenac** p.o.

GPTNERMED: Die **Furosemid** wird abgesetzt.

GPTNERMED: Eine Woche nach der Behandlung kann **Aciclovir** (**400mg** bzw. **1g**) eingesetzt werden.

GPTNERMED: **Kortison** **20mg** **1x** **1x** **1x** **1x**

GPTNERMED: **Bosentan** **62,5** **mg** 2-0-0

GPTNERMED: Häufig kommt es zu **Nebenwirkungen**, wie etwa **Akne** und **Hyperplasie** von Mundsohle.

GPTNERMED: D: Nach 3-stündiger Einwirkung mit **1mg** **Ropivacaïn** besteht eine ausgeprägte, kalte Schwellung am Kopf mit kleiner Ödembildung.

GPTNERMED: Die Klinik des Patienten war sehr unruhig, wobei der Patient eine vermehrte **Atemnot** **50mg** **Nifedipin** **90mg** pro Tag

GPTNERMED: **Budesonid** **2mg** **Methylprednisolon** **500mg** **Prednisolon** **50mg** **Prednisolon** **50mg**

GPTNERMED: Die Patientin bekommt täglich **25mg** **Pantoprazol** p.o.

GPTNERMED: Ich habe **Prednisolon** **60** **mg/d** eingenommen.

Figure 5.9: Samples from the LLM-generated dataset (GPTNERMED). A colored text indicates a present annotation span. The encoded label classes are `Medikation` (green), `Dosis` (red) and `Diagnose` (blue).

- **Instances of Obviously Flawed Annotation**: Closely related to the former aspect, certain annotations can be immediately identified as incorrect. In practice, this mostly applies to the `Diagnose` label class in particular (e.g. "Zervix-PE",

"Nebenwirkung" as diagnosis). Other label classes are affected as well to a lesser extent.

- **Unreliability in Complex Situation**: In more complex cases, the generated annotations are less reliable. For instance, measured blood glucose scores are detected as `Dosis` ("90/50 mmHg") as this particular label class tends to be tied to number appearances rather than to the actual dosage semantics. Other label classes like `Diagnose` may naturally be more difficult to define and identify consistently. For instance, general health issues may not always be considered as diagnoses, leading to potential divergence from the intended notion of diagnosis within the annotated corpus.
- **Effectiveness of Medication Annotation**: Within the shown corpus subset only one drug-related instance could be identified for clearly lacking a proper annotation span ("Mosapride"). In this context, it is worth noting that the synthesized corpus appears to contain a *diverse* set of medication names that actually refer to *valid* drug names.

NER Inspection The NER models are further applied to the text from the OoD dataset. The first samples from the dataset along with the ground truth annotation and the respective annotation from the individual NER models are provided in Figure 5.10.

The visualized results from the NER models largely confirm the quantitative findings from the NER models as well as certain artifacts reported in the qualitative corpus analysis. The major findings can be summarized as follows:

- **Precision and Recall Rates**: The GBERT model tends to prioritize recall over precision, in particular when compared to the GottBERT model.
- **Model Robustness**: The GBERT and GottBERT models appear less erratic, and thus, more robust compared to the German MedBERT model. The German MedBERT model occasionally confuses the label class `Medikation` with `Diagnose` or `Dosis`. Although in limited cases, GBERT and GottBERT also detect relevant instances using an incorrect label class.
- **Variability in Reliability Across Label Classes**: Among the different label classes, `Medikation` appears to be detected more reliably than others, especially `Dosis`. For the `Dosis` label, the models tend to associate predictions with spans containing explicit numerical characters, rather than relying on dosage or strength-specific terminology. Naturally, a major contributing factor may be traced back to flaws originating from the synthesized dataset used for training, which itself struggles to assign correct annotations in certain contexts.
- **Fringe Cases**: Due to complexity and potential subjectivity of certain label classes and situational contexts, incorrect predictions of certain NER models may be perceived debatable or reasonable to some extent. However, this complexity cannot be reflected by the measured precision, recall and F1-scores.

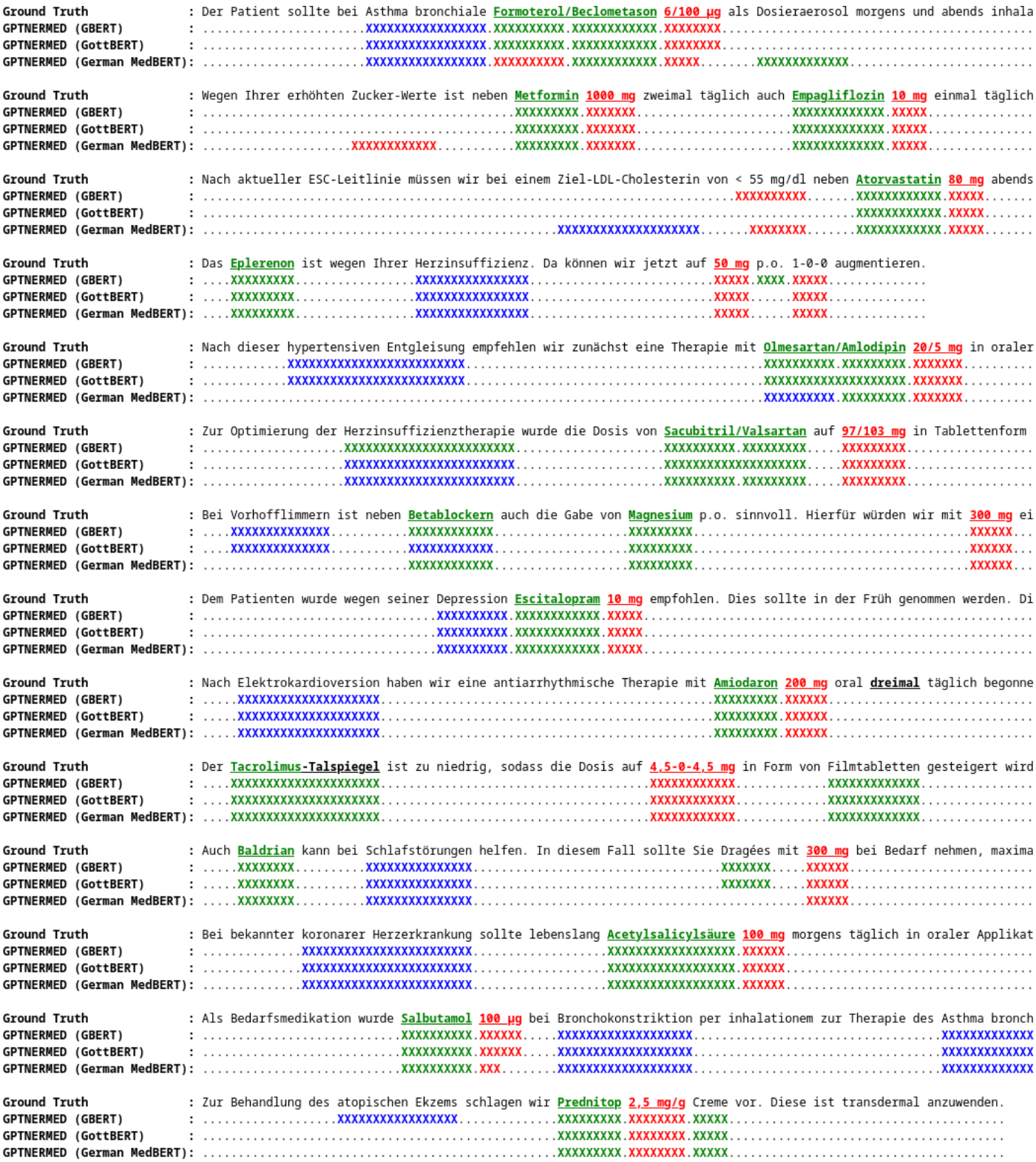

Figure 5.10: Samples from the out-of-distribution (OoD) set. A colored text indicates a present annotation span from the ground truth. The underlined text indicates the span aligned onto the less fine-grained token structure used for evaluation and training. The tokenization is performed by the German tokenizer from Spacy. The colored `X` characters designate predicted annotation spans. For the ground truth, the encoded label classes are `Drug` (green) and `Strength` (red). The predicted label classes are `Medikation` (green), `Dosis` (red) and `Diagnose` (blue). The text is cut off on the right side.

On a general note, it should be considered that the presented results are obtained while neither the NER models nor the generated dataset have been provided with explicit definitions on label classes. Annotation issues that may occur in ambiguous or debatable contexts can only be addressed by agreeing on consistent and clear label definitions. Since such label definitions may be agreed on differently, depending on the actual use case and its task-specific sub-domain, generically designed NER models inherently face this issue.

5.4 Named Entity Recognition on Weakly-annotated, WikiData-defined Corpus

Within the context of this study, a single label class NER with the focus on medication detection is investigated, which also facilitates the evaluation on external datasets.

The experiments are based on specific German Wikipedia[5] (February 22th, 2024) and WikiData[6] (February 23th, 2024) dumps as fixed experiment resources. However, prior to the creation of datasets dedicated to a given SPARQL query, such SPARQL query needs to be evaluated and yield a set of explicit WikiData entities. Although this step could conceptually be performed exclusively with a WikiData dump by loading the dump resource into a compatible graph database, the loading procedure is not well supported and computationally expensive. Therefore, the official WikiData SPARQL API endpoint[7] is utilized instead (queried on April 17th, 2024). Because the state of the WikiData graph is subject to daily changes, identical SPARQL queries may yield different responses over time, and thus, also the obtained datasets may differ when the experiments are repeated.

5.4.1 Dataset Crafting and Imputation

As an initial step, the SPARQL query that defines the entity class and its associated concepts indexed by WikiData is specified. For the objective of describing the medication-related entity class, an entity is considered a medication in case of the presence of an ATC code assigned to the WikiData entity via the relation *P267* ("ATC code"). The corresponding SPARQL query therefore is used to obtain all WikiData entities with an assigned ATC code.

```
SELECT ?item WHERE
{
    ?item wdt:P267 ?atccode .
}
```

The query yields a result set of 4,430 WikiData entities as of October 2nd, 2024[8].

The raw dataset is used to train several custom NER models using an adaptive loss scaling mechanism. The implementation is based on the Huggingface Transformers library in version 4.36.2. As pre-trained encoder model, the generic GottBERT model is used. The NER action labels follow the IOB2 scheme and are predicted by an NER head as a single, randomly initialized linear layer. The action decoding uses a simple

[5] https://dumps.wikimedia.org/dewiki/latest/dewiki-latest-pages-meta-current.xml.bz2 (accessed Februrary 22th, 2024)

[6] https://dumps.wikimedia.org/wikidatawiki/entities/latest-all.json.bz2 (accessed February 22th, 2024)

[7] https://query.wikidata.org/sparql (accessed April 17th, 2024)

[8] https://query.wikidata.org/#SELECT%20%28COUNT%28%2a%29%20AS%20%3Fcount%29%0AWHERE%20%7B%0A%20%20%20%20%3Fitem%20wdt%3AP267%20%3Fatccode%20.%0A%7D (accessed October 2nd, 2024)

greedy decoding strategy using structured prediction as decoding mechanism to filter out invalid actions.

The training parameters are mostly kept to their default values with only minor changes along with the Adam optimizer (learning rate = 5×10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$). The raw dataset is split into a training and evaluation set part (90%) and a test set part (10%). The joint training and evaluation set is further split into training set (80%) and evaluation set (20%).

The token loss scaling factors are set to $\omega_{pos} = 1.337$ for positive and $\omega_{neg} = 0.748$ for negative token positions according to the loss scaling formulas to counterbalance the different frequencies of negative and positive tokens. As the scaling factor for unknown token positions is choosable as a hyperparameter, multiple runs are performed with different settings for the ω_{unk} hyperparameter, namely $\omega_{unk} \in \{1.0, 0.8, 0.5, 0.2, 0.1, 0.05, 0.01\}$. The individual training took between 2:00h and 2:10h for three epochs on a single NVIDIA Titan RTX GPU. The best model checkpoints are chosen based on the lowest loss on the evaluation set.

After the training, the models are loaded and re-applied to the initial raw dataset in inference mode to impute the weak annotation and generate the synthetic, yet fully-annotated datasets. Therefore, multiple datasets with different annotations are obtained. The basic statistics on the raw corpus as well as on the synthetic corpora for different ω_{unk} settings are shown in Table 5.11.

Property	**Value**
#Samples	84,478
#Sents	90,918
#Tokens	2,023,187
#Labels (pos), raw	105,207
#Labels (neg), raw	145,492
#Labels (pos), ω_{unk} = 0.01	135,795
#Labels (pos), ω_{unk} = 0.05	127,950
#Labels (pos), ω_{unk} = 0.1	128,157
#Labels (pos), ω_{unk} = 0.2	120,973
#Labels (pos), ω_{unk} = 0.5	114,339
#Labels (pos), ω_{unk} = 0.8	110,237
#Labels (pos), ω_{unk} = 1.0	110,024

Table 5.11: Statistics on the obtained datasets for different ω_{unk} settings. The corpora are based on the SPARQL query that identifies all WikiData entities with assigned ATC codes.

Given the statistics, the impact of the choosable hyperparameter ω_{unk} is clearly observable. As the scaling factor decreases, the number of predicted entities increases monotonically, except for the $\omega_{unk} = 0.05$ setting.

5.4.2 NER Training

All corpora acquired under different ω_{unk} hyperparameter settings are used to train a corresponding NER model following a classical approach that assumes the training data to behave like a fully-annotated corpus. Therefore, the NER pipeline of Spacy (version 3.7.4) is used with its default hyperparameter configuration (learning rate = 5×10^{-5}, Adam optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, 250 warmup steps), in combination with the pre-trained transformer model *GerMedBERT's medBERT.de* as trainable encoder block. For training, the dataset is once again split into a training and evaluation set part (90%) and a test set part (10%). The joint training and evaluation set is further split into training set (80%) and evaluation set (20%). The training time took 58–130 minutes per session on a single NVIDIA Titan RTX GPU. The best models are chosen based on the best exact entity-wise F1-score on the evaluation set.

5.4.3 Evaluation

Quantitative Analysis

The final NER models are evaluated on several external datasets and the OoD set, partially introduced as result materials within this work (GERNERMED++ and GPT-NERMED). The evaluation scores are computed using an entity-wise evaluation setup with lenient span matching. The obtained scores are shown in Table 5.12 for various ω_{unk} settings. The *BratEval*[9] code is used in *overlap* mode to evaluate the lenient scores.

Without further interpretation, a general trend can be observed in the ratio between the precision and recall scores in correlation with the applied ω_{unk} value. While decreasing ω_{unk} values can yield minor drops in the precision scores, the recall scores appear to drastically improve. In essence, it indicates that the measure can lead to a more balanced precision and recall score ratio as well as to an overall improved F1-score. This observation, in particular with respect to the gains in recall scores, aligns well with the effect shown in Table 5.11 that a lower ω_{unk} scaling value tends to lead to more annotated entities assigned to the corpus text. However, the datasets do not behave identical and the NER models perform best on the datasets with different ω_{unk} scaling hyperparameters according to the scores.

Qualitative Analysis

As the whole process involves both an initial yet highly customized NER training for the annotation imputation, as well as a subsequent conventional NER training on the imputed dataset, both steps are investigated individually using a sample-wise analysis.

Dataset Inspection A randomly sampled snapshot from the datasets is given in Figure 5.11, highlighting the actual effects on the resulting datasets for different ω_{unk} values. To accompany the samples with the ATC-related information from WikiData, Table 5.13 is provided.

[9] https://github.com/READ-BioMed/brateval/tree/v0.3.2 (accessed October 4th, 2024)

ω_{unk}	Dataset	Precision	Recall	F1-score
0.01	OoD [Drug]	.8857	.8611	**.8732**
0.05		.8611	.8611	.8611
0.1		.8205	.8889	.8533
0.2		.8857	.5611	**.8732**
0.5		.8611	.8611	.8611
0.8		.8857	.8611	**.8732**
1.0		.8485	.7778	.8116
0.01	Bronco150 [MEDICATION]	.8103	.7505	**.7792**
0.05		.8209	.7302	.7729
0.1		.7762	.7727	.7745
0.2		.8014	.7538	.7768
0.5		.8381	.6728	.7464
0.8		.8618	.5865	.6980
1.0		.8537	.5983	.7035
0.01	GERNERMED++ [Drug]	.8104	.7897	**.7999**
0.05		.8363	.7383	.7843
0.1		.7627	.7654	.7640
0.2		.8453	.7526	.7963
0.5		.8518	.7152	.7776
0.8		.8773	.6876	.7710
1.0		.8831	.6841	.7710
0.01	GPTNERMED [Medikation]	.8002	.8802	.8383
0.05		.8286	.8638	.8458
0.1		.7410	.8787	.8040
0.2		.8553	.8537	**.8545**
0.5		.8300	.8413	.8356
0.8		.8145	.8151	.8148
1.0		.8336	.8172	.8253
0.01	CARDIO:DE [DRUG, ACTIVEING]	.5402	.7266	.6197
0.05		.5219	.6774	.5895
0.1		.5786	.7278	**.6447**
0.2		.5352	.7107	.6106
0.5		.5856	.6506	.6163
0.8		.6262	.5731	.5985
1.0		.5634	.5924	.5775
0.01	GGPOnc 2 [Clinical_Drug] (short, fine)	.1908	.7257	.3021
0.05		.2386	.6944	**.3552**
0.1		.1855	.7026	.2935
0.2		.2324	.6635	.3442
0.5		.2085	.6265	.3128
0.8		.2143	.5805	.3131
1.0		.2425	.5702	.3402

Table 5.12: NER scores on external and the out-of-distribution (OoD) datasets. The experiments are conducted using BratEval in *overlap* mode for all different ω_{unk} values. The harmonized label classes are given in square brackets.

```
Raw     : aut, dass sie mit zwei Donor-Atomen (zum Beispiel Phosphor, Stickstoff oder Schwefel) und einer σ-Bindung an das (Übergangs-)Metall bin
ATC 0.01: aut, dass sie mit zwei Donor-Atomen (zum Beispiel Phosphor, Stickstoff oder Schwefel) und einer σ-Bindung an das (Übergangs-)Metall bin
ATC 0.05: aut, dass sie mit zwei Donor-Atomen (zum Beispiel Phosphor, Stickstoff oder Schwefel) und einer σ-Bindung an das (Übergangs-)Metall bin
ATC 0.1 : aut, dass sie mit zwei Donor-Atomen (zum Beispiel Phosphor, Stickstoff oder Schwefel) und einer σ-Bindung an das (Übergangs-)Metall bin
ATC 0.2 : aut, dass sie mit zwei Donor-Atomen (zum Beispiel Phosphor, Stickstoff oder Schwefel) und einer σ-Bindung an das (Übergangs-)Metall bin
ATC 0.5 : aut, dass sie mit zwei Donor-Atomen (zum Beispiel Phosphor, Stickstoff oder Schwefel) und einer σ-Bindung an das (Übergangs-)Metall bin
ATC 0.8 : aut, dass sie mit zwei Donor-Atomen (zum Beispiel Phosphor, Stickstoff oder Schwefel) und einer σ-Bindung an das (Übergangs-)Metall bin
ATC 1.0 : aut, dass sie mit zwei Donor-Atomen (zum Beispiel Phosphor, Stickstoff oder Schwefel) und einer σ-Bindung an das (Übergangs-)Metall bin

Raw     : tes selbst nicht mehr produzieren kann, etwa eine bestimmte Aminosäure.
ATC 0.01: tes selbst nicht mehr produzieren kann, etwa eine bestimmte Aminosäure.
ATC 0.05: tes selbst nicht mehr produzieren kann, etwa eine bestimmte Aminosäure.
ATC 0.1 : tes selbst nicht mehr produzieren kann, etwa eine bestimmte Aminosäure.
ATC 0.2 : tes selbst nicht mehr produzieren kann, etwa eine bestimmte Aminosäure.
ATC 0.5 : tes selbst nicht mehr produzieren kann, etwa eine bestimmte Aminosäure.
ATC 0.8 : tes selbst nicht mehr produzieren kann, etwa eine bestimmte Aminosäure.
ATC 1.0 : tes selbst nicht mehr produzieren kann, etwa eine bestimmte Aminosäure.

Raw     : Vitamin-D-Metaboliten (Calcitriol, Alfacalcidol, Paricalcitol) und Calcimimetica (Cinacalcet) kann ein überschießender Knochenumbau (Os
ATC 0.01: Vitamin-D-Metaboliten (Calcitriol, Alfacalcidol, Paricalcitol) und Calcimimetica (Cinacalcet) kann ein überschießender Knochenumbau (Os
ATC 0.05: Vitamin-D-Metaboliten (Calcitriol, Alfacalcidol, Paricalcitol) und Calcimimetica (Cinacalcet) kann ein überschießender Knochenumbau (Os
ATC 0.1 : Vitamin-D-Metaboliten (Calcitriol, Alfacalcidol, Paricalcitol) und Calcimimetica (Cinacalcet) kann ein überschießender Knochenumbau (Os
ATC 0.2 : Vitamin-D-Metaboliten (Calcitriol, Alfacalcidol, Paricalcitol) und Calcimimetica (Cinacalcet) kann ein überschießender Knochenumbau (Os
ATC 0.5 : Vitamin-D-Metaboliten (Calcitriol, Alfacalcidol, Paricalcitol) und Calcimimetica (Cinacalcet) kann ein überschießender Knochenumbau (Os
ATC 0.8 : Vitamin-D-Metaboliten (Calcitriol, Alfacalcidol, Paricalcitol) und Calcimimetica (Cinacalcet) kann ein überschießender Knochenumbau (Os
ATC 1.0 : Vitamin-D-Metaboliten (Calcitriol, Alfacalcidol, Paricalcitol) und Calcimimetica (Cinacalcet) kann ein überschießender Knochenumbau (Os

Raw     : Auch zur Verabreichung von Glucose- oder Xylose-Lösungen für Resorptionstests und für die Ernährung von Fohlen kann das Legen einer Nas
ATC 0.01: Auch zur Verabreichung von Glucose- oder Xylose-Lösungen für Resorptionstests und für die Ernährung von Fohlen kann das Legen einer Nas
ATC 0.05: Auch zur Verabreichung von Glucose- oder Xylose-Lösungen für Resorptionstests und für die Ernährung von Fohlen kann das Legen einer Nas
ATC 0.1 : Auch zur Verabreichung von Glucose- oder Xylose-Lösungen für Resorptionstests und für die Ernährung von Fohlen kann das Legen einer Nas
ATC 0.2 : Auch zur Verabreichung von Glucose- oder Xylose-Lösungen für Resorptionstests und für die Ernährung von Fohlen kann das Legen einer Nas
ATC 0.5 : Auch zur Verabreichung von Glucose- oder Xylose-Lösungen für Resorptionstests und für die Ernährung von Fohlen kann das Legen einer Nas
ATC 0.8 : Auch zur Verabreichung von Glucose- oder Xylose-Lösungen für Resorptionstests und für die Ernährung von Fohlen kann das Legen einer Nas
ATC 1.0 : Auch zur Verabreichung von Glucose- oder Xylose-Lösungen für Resorptionstests und für die Ernährung von Fohlen kann das Legen einer Nas

Raw     :  der Schleimhäute unter Bildung von hypochloriger Säure und Chlorwasserstoffsäure.
ATC 0.01:  der Schleimhäute unter Bildung von hypochloriger Säure und Chlorwasserstoffsäure.
ATC 0.05:  der Schleimhäute unter Bildung von hypochloriger Säure und Chlorwasserstoffsäure.
ATC 0.1 :  der Schleimhäute unter Bildung von hypochloriger Säure und Chlorwasserstoffsäure.
ATC 0.2 :  der Schleimhäute unter Bildung von hypochloriger Säure und Chlorwasserstoffsäure.
ATC 0.5 :  der Schleimhäute unter Bildung von hypochloriger Säure und Chlorwasserstoffsäure.
ATC 0.8 :  der Schleimhäute unter Bildung von hypochloriger Säure und Chlorwasserstoffsäure.
ATC 1.0 :  der Schleimhäute unter Bildung von hypochloriger Säure und Chlorwasserstoffsäure.

Raw     : Es ist – ähnlich dem Adrenalin und Noradrenalin – ein Sympathomimetikum, also ein Stoff, der die Wirkungen des sympathischen Nervensyst
ATC 0.01: Es ist – ähnlich dem Adrenalin und Noradrenalin – ein Sympathomimetikum, also ein Stoff, der die Wirkungen des sympathischen Nervensyst
ATC 0.05: Es ist – ähnlich dem Adrenalin und Noradrenalin – ein Sympathomimetikum, also ein Stoff, der die Wirkungen des sympathischen Nervensyst
ATC 0.1 : Es ist – ähnlich dem Adrenalin und Noradrenalin – ein Sympathomimetikum, also ein Stoff, der die Wirkungen des sympathischen Nervensyst
ATC 0.2 : Es ist – ähnlich dem Adrenalin und Noradrenalin – ein Sympathomimetikum, also ein Stoff, der die Wirkungen des sympathischen Nervensyst
ATC 0.5 : Es ist – ähnlich dem Adrenalin und Noradrenalin – ein Sympathomimetikum, also ein Stoff, der die Wirkungen des sympathischen Nervensyst
ATC 0.8 : Es ist – ähnlich dem Adrenalin und Noradrenalin – ein Sympathomimetikum, also ein Stoff, der die Wirkungen des sympathischen Nervensyst
ATC 1.0 : Es ist – ähnlich dem Adrenalin und Noradrenalin – ein Sympathomimetikum, also ein Stoff, der die Wirkungen des sympathischen Nervensyst

Raw     : nd die vier gebräuchlichsten Wirkstoffe gegen Kopfschmerzen Acetylsalicylsäure, Paracetamol, Ibuprofen sowie Propyphenazon. Auf Grund d
ATC 0.01: nd die vier gebräuchlichsten Wirkstoffe gegen Kopfschmerzen Acetylsalicylsäure, Paracetamol, Ibuprofen sowie Propyphenazon. Auf Grund d
ATC 0.05: nd die vier gebräuchlichsten Wirkstoffe gegen Kopfschmerzen Acetylsalicylsäure, Paracetamol, Ibuprofen sowie Propyphenazon. Auf Grund d
ATC 0.1 : nd die vier gebräuchlichsten Wirkstoffe gegen Kopfschmerzen Acetylsalicylsäure, Paracetamol, Ibuprofen sowie Propyphenazon. Auf Grund d
ATC 0.2 : nd die vier gebräuchlichsten Wirkstoffe gegen Kopfschmerzen Acetylsalicylsäure, Paracetamol, Ibuprofen sowie Propyphenazon. Auf Grund d
ATC 0.5 : nd die vier gebräuchlichsten Wirkstoffe gegen Kopfschmerzen Acetylsalicylsäure, Paracetamol, Ibuprofen sowie Propyphenazon. Auf Grund d
ATC 0.8 : nd die vier gebräuchlichsten Wirkstoffe gegen Kopfschmerzen Acetylsalicylsäure, Paracetamol, Ibuprofen sowie Propyphenazon. Auf Grund d
ATC 1.0 : nd die vier gebräuchlichsten Wirkstoffe gegen Kopfschmerzen Acetylsalicylsäure, Paracetamol, Ibuprofen sowie Propyphenazon. Auf Grund d

Raw     : en Polysaccharide, einfache Kohlenhydrate, einige Alkohole, Aminosäuren, Buttersäure und andere organische Säuren).
ATC 0.01: en Polysaccharide, einfache Kohlenhydrate, einige Alkohole, Aminosäuren, Buttersäure und andere organische Säuren).
ATC 0.05: en Polysaccharide, einfache Kohlenhydrate, einige Alkohole, Aminosäuren, Buttersäure und andere organische Säuren).
ATC 0.1 : en Polysaccharide, einfache Kohlenhydrate, einige Alkohole, Aminosäuren, Buttersäure und andere organische Säuren).
ATC 0.2 : en Polysaccharide, einfache Kohlenhydrate, einige Alkohole, Aminosäuren, Buttersäure und andere organische Säuren).
ATC 0.5 : en Polysaccharide, einfache Kohlenhydrate, einige Alkohole, Aminosäuren, Buttersäure und andere organische Säuren).
ATC 0.8 : en Polysaccharide, einfache Kohlenhydrate, einige Alkohole, Aminosäuren, Buttersäure und andere organische Säuren).
ATC 1.0 : en Polysaccharide, einfache Kohlenhydrate, einige Alkohole, Aminosäuren, Buttersäure und andere organische Säuren).
```

Figure 5.11: Samples from the raw and imputed Wikipedia-sourced corpora. The colored text indicates a present annotation span from the raw (weakly-annotated) and imputed datasets. The underlined text indicates the span aligned onto the less fine-grained token structure used for evaluation and training. The tokenization is performed by the German tokenizer from Spacy. The imputed datasets stem from the custom NER models trained with a specified ω_{unk} value. The text is cut off on the right side, and cropped to begin at most 40 characters before the first raw ATC annotation.

Sample Nr.	Term	WikiData Item	ATC Code
1	Phosphor	Q674	*none*
	Stickstoff	Q627	V03AN04
	Schwefel	Q682	*none*
2	Aminosäure	Q8066	B05BA01
3	Vitamin D	Q175621	A11CC
	Calcitriol	Q139195	D05AX03/A11CC04
	Alfacalcidol	Q155883	A11CC03
	Paricalcitol	Q155746	H05BX02
	Calcimimetica	Q5018775	*none*
	Cinacalcet	Q193978	H05BX01
4	Glucose	Q37525	V06DC01/B05CX01/V04CA02
	Xylose	Q66589603	*none*
5	hypochlorige Säure	Q407318	*none*
	Chlorwasserstoffsäure	Q2409	B05XA13/A09AB03
6	Adrenalin	Q132621	R03AA01/C01CA24/A01AD01/ R01AA14/B02BC09/S01EA01
	Noradrenalin	Q186242	C01CA03
	Sympathomimetikum	Q598405	R01BA
7	Acetylsalicylsäure	Q18216	A01AD05/B01AC06/N02BA01
	Paracetamol	Q57055	N02BE01
	Ibuprofen	Q598405	C01EB16/G02CC01/M01AE01/ M02AA13/R02AX02
	Propyphenazon	Q425111	N02BB04
8	Polysaccharide	Q134219	*none*
	Kohlenhydrate	Q11358	*none*
	Alkohole	Q156 (“alcohols”)	*none*
		Q153 (“ethanol”)	V03AZ01/V03AB16/D08AAX08
		Q154 (“alcoholic beverage”)	*none*
	Aminosäure	Q8066	B05BA01
	Buttersäure	Q193213	*none*
	organische Säure	Q421948	*none*

Table 5.13: Terms and their associated WikiData and ATC groundings. The terms and sample numbers correspond to the samples from Figure 5.11, counting from top (1) to bottom (8). The ATC codes are extracted from the WikiData items (accessed October 7th, 2024).

A few aspects may be highlighted from the text samples:

- **Loss-scaling Effect**: The effect of the token-wise loss scaling can be seen in a few instances and its effect to mitigate the weak annotation style appears to happen for lower ω_{unk} values (e.g. “Vitamin D”, “Paricalcitol”, “Calcimimetica”).
- **Adherence to ATC Code**: In general, the annotations tend to adhere to the presence or lack of ATC codes assigned to the mentioned concepts. It occurs despite the fact that other terms may be assumed to have similar characteristics (namely, an ATC code assigned) due to certain sentence structures (e.g. concept enumerations).

- **Arguable Edge Cases**: The correct annotation in certain instances are not obvious, since a mention could be interpreted to ambiguously refer to various WikiData items, partially with an ATC code assigned, and thus, a final decision may be unclear (e.g. "Alkohole"). The mention "Calcimimetica" could be perceived as an arguable annotation sample as well since it refers to a medication instance. Yet due to the fact that its WikiData item has no ATC code assigned and only ATC-assigned entities are extracted, a potentially predicted annotation remains incorrect.
- **Formal Text Style**: As Wikipedia's text style is largely written in a rather formal way, the extracted text samples reflect this notion. In this regard, the text is clearly biased towards its distinct style.

NER Inspection The imputed datasets, that are serving as training data, determine the behavior of the final NER models. Differences in their behavior patterns are evaluated sample-wise on the OoD corpus. The predicted annotations for the first OoD samples are shown in Figure 5.12. The associated WikiData-based ATC information is shown in Table 5.14.

The visualization mostly highlights the overall effectiveness of the approach on the OoD dataset, as most medication-related items are correctly identified in the given example. Only minor observations can be drawn from the example. First, in the $\omega_{unk} = 1.0$ setup, the NER model may behave more conservatively with regards to the detected spans, leading to a missed detection of a span ("Atorvastarin"), yet this effect is already evidently shown in its lower recall rate in Table 5.12. Second, instances like "Magnesium" and "Sacubitril" may be associated with WikiData entities without any ATC codes assigned, and thus, is not supposed to be detected by the ATC-focused NER models. Even though a dedicated WikiData entity titled "Sacubitril/Valsaratan" is present in the WikiData database and has an ATC code assigned, this entity has no dedicated *German* Wikipedia page and is therefore not covered by the current ATC-based NER models.

The NER performance of the NER models suffers in particular on the CARDIO:DE and GGPOnc 2 corpora. A further, manual inspection reveals the following main observations:

- **Ground Truth Inconsistencies**: Occasionally, certain annotation instances may be intrinsically inconsistent or even contradictory, and some phrases may simply be overlooked by the annotators. An example of an incoherent annotation of the concept "contrast agent" from the GGPOnc 2.0 corpus is shown in Figure 5.14. Regarding missed annotation phrases, one example from the CARDIO:DE corpus is shown in Figure 5.13 (bottom sample).
- **Entity Sparsity Issue**: The GGPOnc 2.0 corpus consists of many individual and comparatively shorter text samples (about 86k samples), unlike, for example, Bronco150, which consists of five rather large text samples. As a result, a high

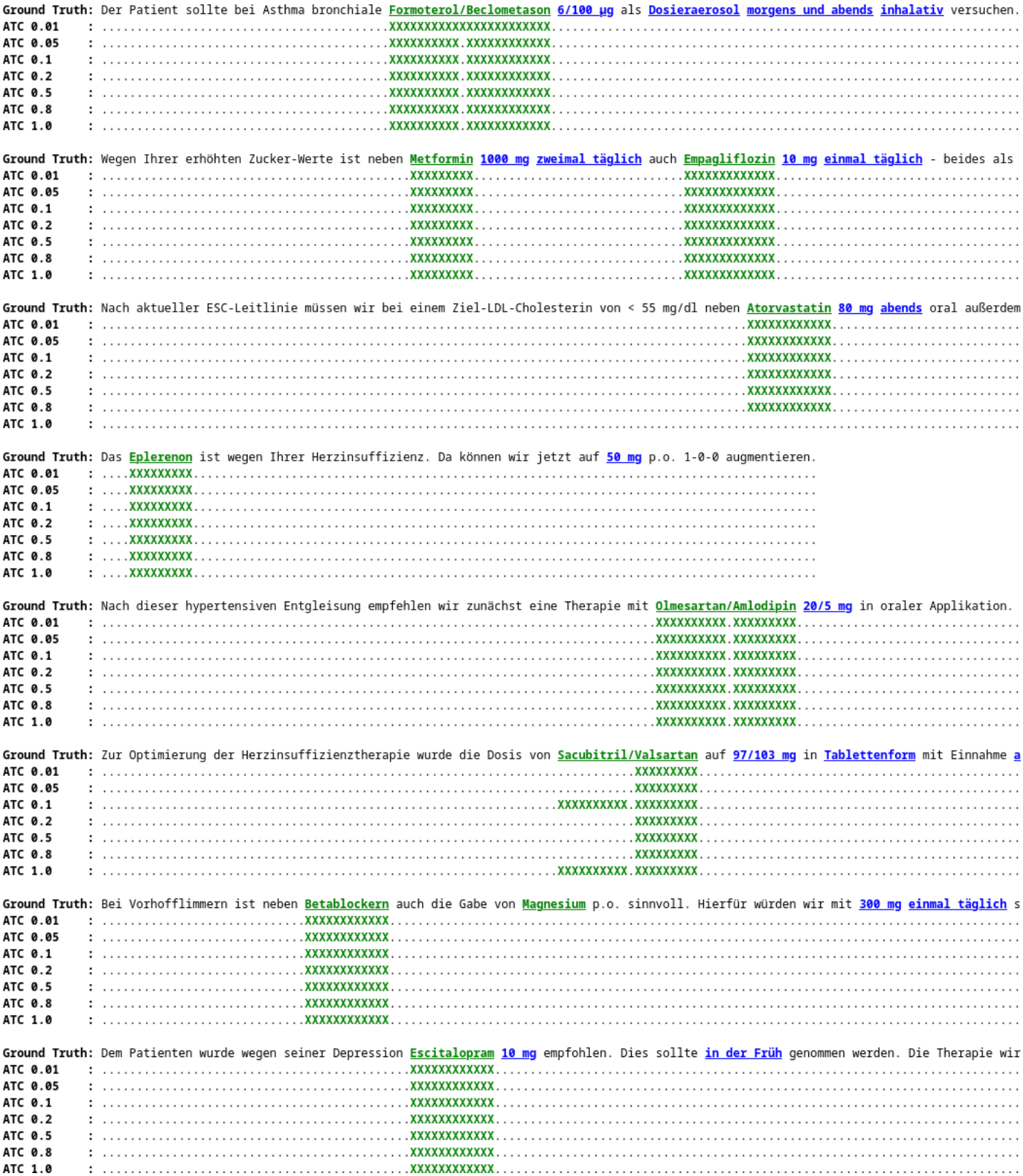

Figure 5.12: Samples from the out-of-distribution (OoD) set. A colored text indicates a present annotation span from the ground truth. The underlined text indicates the span aligned onto the less fine-grained token structure used for evaluation and training. The tokenization is performed by the German tokenizer from Spacy. The colored `X` characters designate predicted annotation spans from the NER model trained given the specified ω_{unk} value. For the ground truth, the encoded label classes are `Drug` (green) and all remaining label classes (blue). The single, predicted label class is `ATC_CODE` (green). The text is cut off on the right side.

Sample Nr.	Term	WikiData Item	ATC Code
1	Formoterol	Q637247	R03AC13
	Beclometason	Q421475	R01AD01
2	Metformin	Q19484	A10BA02
	Empagliflorin	Q5373824	A10BX12/A10BK03
3	Atorvastatin	Q668093	C10AA05
4	Eplerenon	Q423804	C03DA04
5	Olmesartan	Q421156	C09CA08
	Amlopidin	Q411347	C09DX04
6	Sacubitril	Q17811448	*none*
	Valsartan	Q155472	C09CA03
	Sacubitril/Valsartan	Q6457467	C09DX04
7	Betablocker	Q816759	C07
	Magnesium	Q660 (“magnesium”)	*none*
		Q288266 (“magnesium sulfate”)	D11AX05/V04CC02/B05XA05/ A06AD04/A12C02
		Q6731379 (“magnesium aspartate”)	A12CC05
8	Escitalopram	Q423757	N06AB10

Table 5.14: Terms and their associated WikiData and ATC groundings. The terms and sample numbers correspond to the samples from Figure 5.12, counting from top (1) to bottom (8). The ATC codes are extracted from the WikiData items (accessed October 7th, 2024).

percentage of these GGPOnc 2.0 samples do not contain medication-related annotated text phrases at all. The CARDIO:DE corpus has another, slightly different characteristic. It contains rather long text samples, where large parts of the text do not contain any medication-related phrases. Both described properties can be summarized as a form of entity sparsity. Entity sparsity can be quantified by measuring the relative frequency of medication-related tokens and the relative frequency of empty samples, referring to samples that do not contain any medication-related annotation at all. The quantified statistics are given in Table 5.15. In entity sparsity contexts, exemplified in Figure 5.14 (bottom sample), the NER models are susceptible for hallucinations and may erroneously detect false positive entities. Interestingly, these hallucations from the NER models could be mitigated by appending an additional, medication phrase-containing sentence (“Mittels Cortison behandelbar.”) to the sample (”- Weiße oder rote Flecken auf der Mundschleimhaut an jeglicher Lokalisation“), causing the models to only detect the newly added “Cortison” medication mention. This hints at a potential issue in the training data. Because of the way the training corpus is constructed, the NER models are never trained on samples that contain no medication-related entities, leading to unstable performance on such empty cases.

- **Unnatural Tokens, Codes, and Abbreviations**: Unnatural characters or phrases, partially emerging from the pseudonymization masks, may in some instances lead to an incorrect entity detection, as shown in Figure 5.13 (e.g. top sample) for the

CARDIO:DE dataset ("-UNK-"). Also, certain coding schemes and abbreviations tend to trigger false positive/negative predictions, especially in, but not limited to, entity-sparse contexts.

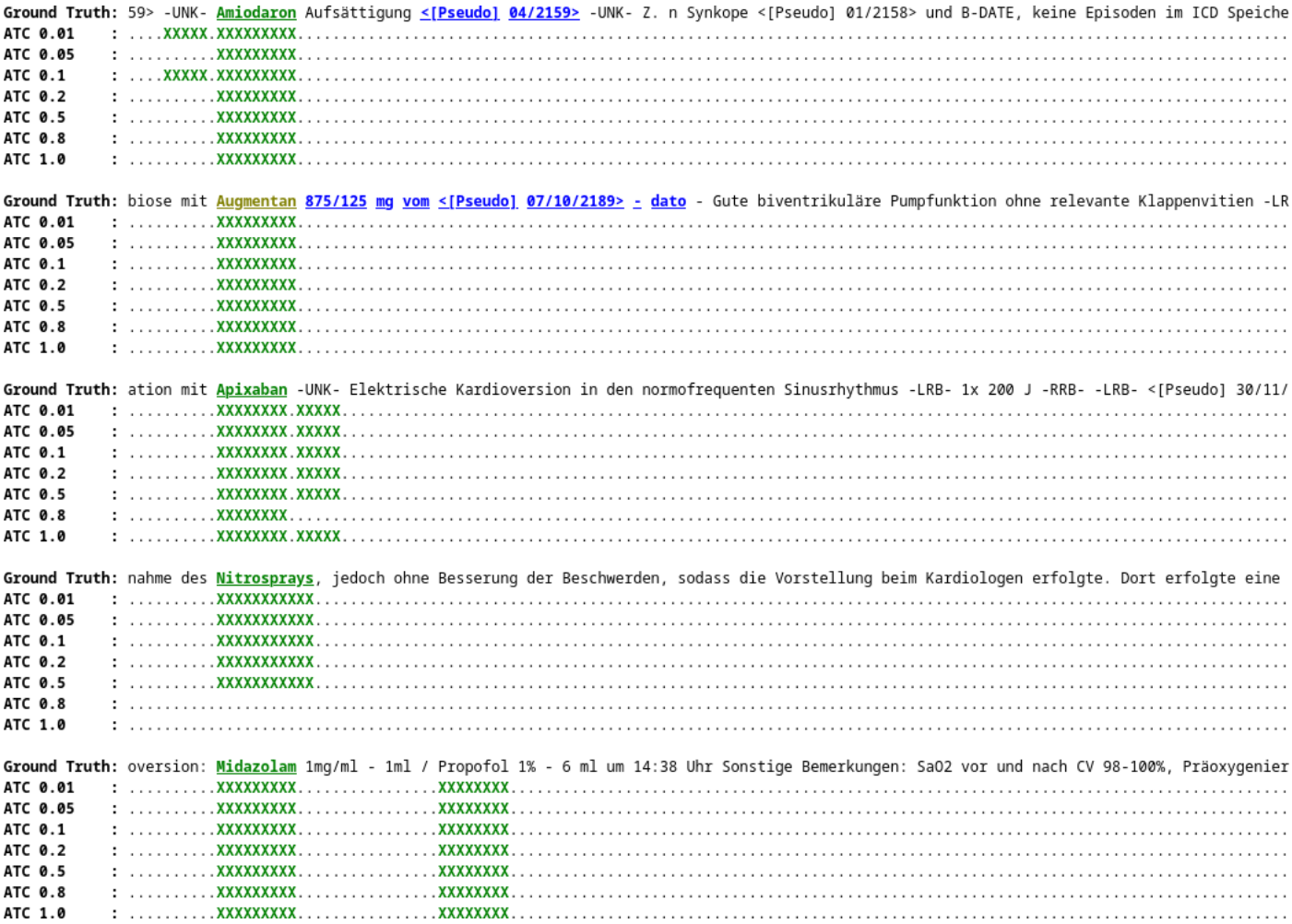

Figure 5.13: Selected samples from CARDIO:DE. The bottom sample shows an instance of a missed medication annotation in the ground truth data. The other samples serve demonstration purposes, including the annotation issues with unnatural text phrases ("-UNK-"). A colored text indicates a present annotation span from the ground truth. The underlined text indicates the span aligned onto the less fine-grained token structure used for evaluation and training. The tokenization is performed by the German tokenizer from Spacy. The colored `X` characters designate predicted annotation spans from the NER model trained given the specified ω_{unk} value. For the ground truth, the encoded label classes are `DRUG` (green) and `ACTIVEING` (olive green) and all remaining label classes (blue). The single, predicted label class is `ATC_CODE` (green). The text is cut off on the right side, and cropped to begin at most 10 characters before the first medication-related ground truth annotation.

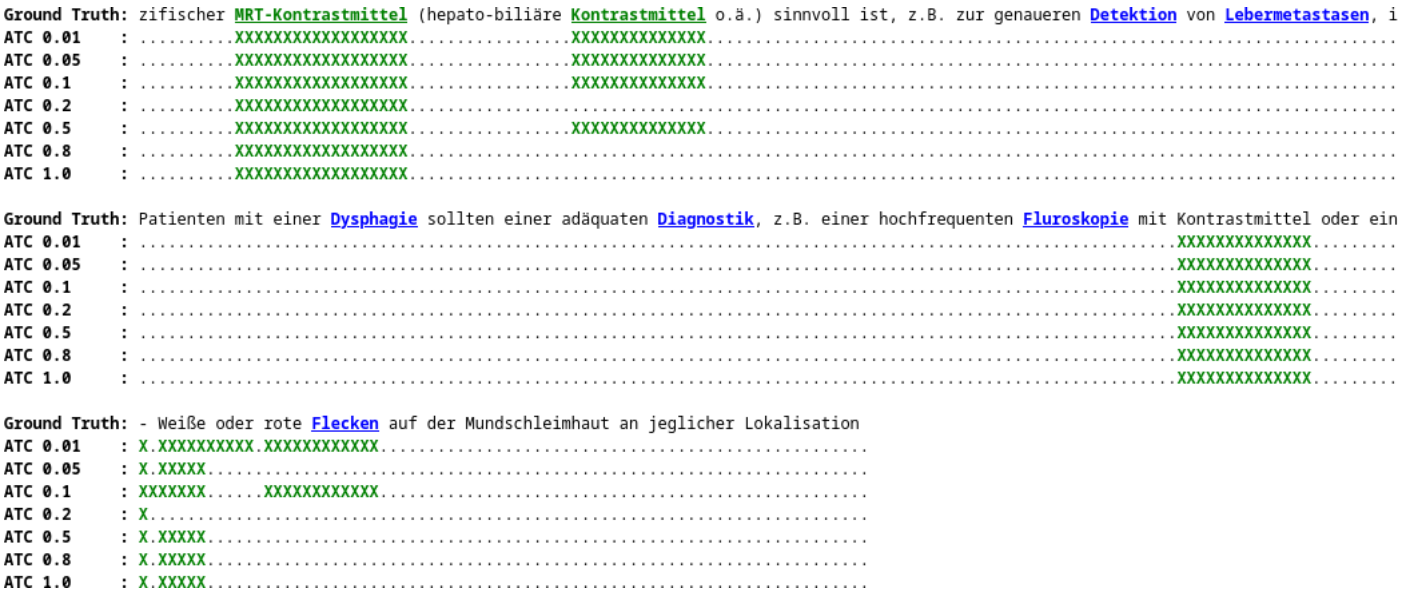

Figure 5.14: Selected samples from GGPOnc 2.0. The bottom sample shows an instance of an empty sample without any medication-related annotation in the ground truth data. The remaining samples (top, middle) show an instance of ground truth inconsistencies ("Kontrastmittel"). A colored text indicates a present annotation span from the ground truth. The underlined text indicates the span aligned onto the less fine-grained token structure used for evaluation and training. The tokenization is performed by the German tokenizer from Spacy. The colored `X` characters designate predicted annotation spans from the NER model trained given the specified ω_{unk} value. For the ground truth, the encoded label classes are `Clinical_Drug` (green) and all remaining label classes (blue). The single, predicted label class is `ATC_CODE` (green). The text is cut off on the right side, and cropped to begin at most 10 characters before the first medication-related ground truth annotation.

Corpus	**Label Class**	**Relative Frequencies** (w.r.t. Label Class) $\frac{\#\text{Label Tokens}}{\#\text{Total Tokens}}$	$\frac{\#\text{Samples}_{\text{w/o Label Tokens}}}{\#\text{Total Samples}}$
CARDIO:DE	`DRUG, ACTIVEING`	0.91%	26.75%
Bronco150	`MEDICATION`	5.47%	0.00%
GGPOnc 2.0 (fine, short)	`Clinical_Drug`	1.03%	87.61%
GGPOnc 2.0 (coarse, short)	`Substance`	1.34%	84.65%
GGPOnc 2.0 (fine, long)	`Clinical_Drug`	1.28%	90.53%
GGPOnc 2.0 (coarse, long)	`Substance`	1.61%	88.23%
GPTNERMED	`Medikation`	8.37%	25.32%
GERNERMED++	`Drug`	7.83%	9.52%
ATC (raw)	`ATC_CODE`	5.47%	0.00%
GSC EMEA	`CHEM`	5.56%	40.00%
GSC Medline	`CHEM`	5.79%	71.00%
GSC Patent	`CHEM`	6.63%	26.00%
OoD	`Drug`	4.78%	3.33%

Table 5.15: Relative frequencies of medication-related tokens and samples. Relative to its total size, the GGPOnc 2.0 and CARDIO:DE corpora feature fewer medication-related tokens compared to the remaining corpora. The GGPOnc 2.0 and GSC Medline corpora show a high percentage of samples without any medication-related annotation. Spacy's German tokenizer is used for the tokenization.

5.5 Quantitative NER Results using Unified Evaluation

The performance scores from the previous NER results are evaluated with different evaluation configurations, and are therefore not directly comparable with each other in a reliable fashion. While the differences in the evaluation strategy stem from the fact that the results for the introduced methodologies are evaluated and reported separately as published research articles, the developed models are again re-evaluated in this section using a unified evaluation strategy.

The evaluation strategy is performed as follows: The ground truth corpora are filtered such that only medication-related entities are kept in the NER annotation entries and renamed into a common, generic label class `Drug`. The remaining entries are further filtered for overlapping spans and only the longest span is kept in these cases. While the overlap removal is not strictly necessary for an entity-wise evaluation, it is motivated by two factors. First, the overlap removal is required for the token-wise evaluation and thus, the filtered ground truth corpora remain comparable. Second, some artifacts are removed from the untouched ground truth corpora. For instance, certain spans are found to occur multiple times in the Bronco150 dataset depending on the number of ontology references that a span is assigned with. Both an exact and a lenient span match strategies are employed. The BratEval tool (version 0.3.2) is used for the evaluation.

The final results are given in Table 5.16 for the key external datasets and in Table 5.17 for the resulting corpora. The results for the external GSC subcorpora are shown in Table 5.18 separately. For a more condensed overview, the pure F1-scores are shown in Table 5.19 (lenient matches) and in Table 5.20 (exact matches).

Even though the unified results offer a reasonably fair comparability across different methods and datasets, it should be emphasized that the ground truth corpora are designed with non-congruent label class definitions and may diverge in text style and appearance.

Drawing conclusions from the results, a clear observation emerges that confirms the superiority of transformer-based encoders over the Token2Vec encoder of Spacy. While two Token2Vec approaches, namely for GERNERMED and GERNERMED++ with Token2Vec encoder, are employed, it is shown in particular for the case of GERNERMED++ that these approaches manage to produce well-performing models on the training dataset but fail to maintain performance across other datasets, indicating issues with the learning of meaningful semantic abstractions as well as the presence of typical overfitting effects. While transformer-based approaches also lose performance on other corpora, their drop in F1-score performance is less severe and the performance remains quite robust when compared to the Token2Vec-based approaches.

The search-based contender, DrNote, which operates on a predefined vocabulary, performs well on a broad set of corpora given the fact that no neural NER methodology is used and the vocabulary is enriched exclusively by crowdsourced WikiData phrases.

Due to the search-based mechanism, the approach tends to achieve higher precision scores over recall scores on several corpora.

Regarding the BERT-based transformers used as embedding encoders, no clear superiority of neither the GermanMedBERT model nor the GottBERT model are evident in the scores for the GPTNERMED approach. However surprisingly in this context, the GottBERT-based model in its base configuration (127M parameters) seems to perform better than its GBERT-based counterpart in its large configuration (337M parameters).

Reviewing the results from a more distant perspective, it appears that the translation- and word alignment-based technique can yield the best training resources and NER model to achieve optimal overall F1-scores across several corpora. The second most successful approach can be attributed to the Wikipedia-sourced, ontology-defined ATC method combined with an applied loss scaling, however it is closely followed by the LLM-based method with its LLM-synthesized corpus.

NER Results		Drug (overlap/lenient)			Drug (exact)		
		Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Bronco150 [Drug = `MEDICATION`]							
Models	DrNote	.6623	.6545	.6583	.5704	.5637	.5670
	GERNERMED	.2179	.4768	.2991	.1675	.3664	.2299
	GERNERMED++ @ German BERT	.7844	.6773	.7270	.7133	.6159	.6611
	GERNERMED++ @ GottBERT	.7498	.7929	.7708	.6905	.7302	.7098
	GERNERMED++ @ Spacy Token2Vec	.3883	.5186	.4441	.3575	.4775	.4088
	GPTNERMED @ GermanMedBERT	.5794	.7198	.6420	.5352	.6649	.5931
	GPTNERMED @ GottBERT	.6886	.8014	.7407	.6403	.7453	.6888
	GPTNERMED @ GBERT	.4681	.9249	.6216	.4340	.8576	.5764
	ATC $\omega_{unk} = 0.01$	.8103	.7505	.7792	.6171	.5715	.5934
	ATC $\omega_{unk} = 0.05$	.8209	.7302	.7729	.6799	.6048	.6402
	ATC $\omega_{unk} = 0.1$	.7762	.7727	.7745	.7100	.7067	.7083
	ATC $\omega_{unk} = 0.2$	.8014	.7538	.7768	.7264	.6832	.7041
	ATC $\omega_{unk} = 0.5$	.8381	.6728	.7464	.7803	.6264	.6949
	ATC $\omega_{unk} = 0.8$	.8618	.5865	.6980	.7956	.5415	.6444
	ATC $\omega_{unk} = 1.0$	.8537	.5983	.7035	.7884	.5526	.6498
CARDIO:DE [Drug = `DRUG`, `ACTIVEING`]							
Models	DrNote	.5612	.6373	.5968	.5439	.6177	.5785
	GERNERMED	.1805	.5853	.2759	.1614	.5232	.2467
	GERNERMED++ @ German BERT	.6678	.7285	.6968	.6429	.7013	.6708
	GERNERMED++ @ GottBERT	.7021	.7944	.7455	.6731	.7616	.7146
	GERNERMED++ @ Spacy Token2Vec	.2641	.5919	.3652	.2601	.5829	.3597
	GPTNERMED @ GermanMedBERT	.4515	.7403	.5609	.4501	.7380	.5591
	GPTNERMED @ GottBERT	.4080	.7720	.5339	.4007	.7582	.5243
	GPTNERMED @ GBERT	.2656	.8647	.4064	.2619	.8525	.4007
	ATC $\omega_{unk} = 0.01$	.5402	.7266	.6197	.4975	.6691	.5707
	ATC $\omega_{unk} = 0.05$	.5219	.6774	.5895	.4918	.6384	.5556
	ATC $\omega_{unk} = 0.1$	.5786	.7278	.6447	.5676	.7140	.6324
	ATC $\omega_{unk} = 0.2$	.5352	.7107	.6106	.5178	.6876	.5908
	ATC $\omega_{unk} = 0.5$	.5856	.6506	.6163	.5827	.6474	.6134
	ATC $\omega_{unk} = 0.8$	.6262	.5731	.5985	.6200	.5674	.5925
	ATC $\omega_{unk} = 1.0$	.5634	.5924	.5775	.5590	.5877	.5730
GGPOnc 1.0 [Drug = `Chemicals_Drugs`]							
Models	DrNote	.8175	.5285	.6420	.7967	.5150	.6256
	GERNERMED	.1248	.2994	.1761	.0678	.1626	.0957
	GERNERMED++ @ German BERT	.6384	.5640	.5989	.5596	.4945	.5250
	GERNERMED++ @ GottBERT	.6372	.5837	.6093	.5607	.5137	.5362
	GERNERMED++ @ Spacy Token2Vec	.2312	.4065	.2948	.1983	.3487	.2528
	GPTNERMED @ GermanMedBERT	.6208	.6417	.6311	.5523	.5708	.5614
	GPTNERMED @ GottBERT	.5247	.7526	.6183	.4483	.6430	.5283
	GPTNERMED @ GBERT	.3630	.8311	.5053	.3118	.7139	.4341
	ATC $\omega_{unk} = 0.01$	.7110	.6154	.6597	.6019	.5209	.5585
	ATC $\omega_{unk} = 0.05$	.7694	.5491	.6408	.6761	.4825	.5631
	ATC $\omega_{unk} = 0.1$	.7222	.5582	.6297	.6483	.5011	.5653
	ATC $\omega_{unk} = 0.2$	.7336	.4397	.5498	.6563	.3934	.4919
	ATC $\omega_{unk} = 0.5$	.6917	.3543	.4686	.6250	.3201	.4234
	ATC $\omega_{unk} = 0.8$	.6799	.2719	.3884	.6303	.2520	.3601
	ATC $\omega_{unk} = 1.0$	.7116	.2757	.3974	.6616	.2563	.3695
GGPOnc 2.0 (fine, short) [Drug = `Clinical_Drug`]							
Models	DrNote	.6834	.5374	.6016	.6334	.4981	.5577
	GERNERMED	.1017	.2992	.1517	.0577	.1699	.0862
	GERNERMED++ @ German BERT	.3306	.7020	.4495	.3039	.6453	.4132
	GERNERMED++ @ GottBERT	.3406	.7293	.4643	.3143	.6730	.4285
	GERNERMED++ @ Spacy Token2Vec	.1639	.4140	.2349	.1561	.3942	.2237
	GPTNERMED @ GermanMedBERT	.3109	.7133	.4330	.2989	.6858	.4164
	GPTNERMED @ GottBERT	.3463	.7502	.4739	.3231	.6998	.4421
	GPTNERMED @ GBERT	.2221	.8562	.3527	.2084	.8033	.3309
	ATC $\omega_{unk} = 0.01$	.1908	.7257	.3021	.1673	.6367	.2650
	ATC $\omega_{unk} = 0.05$	.2386	.6944	.3552	.2248	.6541	.3346
	ATC $\omega_{unk} = 0.1$	.1855	.7026	.2935	.1763	.6676	.2789
	ATC $\omega_{unk} = 0.2$	.2324	.6635	.3442	.2166	.6184	.3209
	ATC $\omega_{unk} = 0.5$	.2085	.6265	.3128	.1993	.5991	.2991
	ATC $\omega_{unk} = 0.8$	.2143	.5805	.3131	.2064	.5591	.3015
	ATC $\omega_{unk} = 1.0$	.2425	.5702	.3402	.2347	.5519	.3293

Table 5.16: Entity-wise NER evaluation on external corpora. BratEval is used for the score computation. Prior to the computation, all unrelated labels are removed and all related labels unified into one, common label class, and a span overlap removal is applied.

NER Results		Drug (overlap/lenient)			Drug (exact)		
		Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
OoD [Drug = `Drug`]							
Models	DrNote	.8667	.7222	.7879	.7667	.6389	.6970
	GERNERMED	.5128	.5556	.5333	.3590	.3889	.3733
	GERNERMED++ @ German BERT	.8537	.9722	.9091	.7805	.8889	.8312
	GERNERMED++ @ GottBERT	.8537	.9722	.9091	.7561	.8611	.8052
	GERNERMED++ @ Spacy Token2Vec	.6750	.7500	.7105	.5500	.6111	.5789
	GPTNERMED @ GermanMedBERT	.7556	.9444	.8395	.6444	.8056	.7160
	GPTNERMED @ GottBERT	.7234	.9444	.8193	.6383	.8333	.7229
	GPTNERMED @ GBERT	.6364	.9722	.7692	.5273	.8056	.6374
	ATC $\omega_{unk} = 0.01$	.8857	.8611	.8732	.7143	.6944	.7042
	ATC $\omega_{unk} = 0.05$	.8611	.8611	.8611	.7222	.7222	.7222
	ATC $\omega_{unk} = 0.1$	.8205	.8889	.8533	.6923	.7500	.7200
	ATC $\omega_{unk} = 0.2$	.8857	.8611	.8732	.7143	.6944	.7042
	ATC $\omega_{unk} = 0.5$	.8611	.8611	.8611	.7222	.7222	.7222
	ATC $\omega_{unk} = 0.8$	.8857	.8611	.8732	.7143	.6944	.7042
	ATC $\omega_{unk} = 1.0$	.8485	.7778	.8116	.6970	.6389	.6667
ATC (raw) [Drug = `ATC_CODE`]							
Models	DrNote	.7198	.7221	.7210	.7067	.7089	.7078
	GERNERMED	.3089	.3255	.3170	.1421	.1498	.1459
	GERNERMED++ @ German BERT	.4709	.7362	.5744	.3830	.5989	.4672
	GERNERMED++ @ GottBERT	.4763	.6997	.5667	.4042	.5938	.4810
	GERNERMED++ @ Spacy Token2Vec	.3639	.3856	.3744	.3126	.3313	.3217
	GPTNERMED @ GermanMedBERT	.4299	.7390	.5436	.3939	.6772	.4981
	GPTNERMED @ GottBERT	.4162	.7270	.5294	.3793	.6626	.4824
	GPTNERMED @ GBERT	.3589	.8578	.5061	.3253	.7774	.4587
	ATC $\omega_{unk} = 0.01$	.7760	.9830	.8673	.6539	.8283	.7308
	ATC $\omega_{unk} = 0.05$	.8210	.9805	.8937	.7302	.8720	.7948
	ATC $\omega_{unk} = 0.1$	.8155	.9802	.8903	.7373	.8862	.8049
	ATC $\omega_{unk} = 0.2$	.8542	.9759	.9110	.7730	.8831	.8244
	ATC $\omega_{unk} = 0.5$	.8889	.9605	.9233	.8090	.8741	.8403
	ATC $\omega_{unk} = 0.8$	.9109	.9507	.9304	.8300	.8663	.8478
	ATC $\omega_{unk} = 1.0$	.9138	.9548	.9338	.8323	.8697	.8506
GERNERMED++ [Drug = `Drug`]							
Models	DrNote	.9086	.5899	.7153	.7712	.5007	.6072
	GERNERMED	.8735	.6357	.7359	.6130	.4461	.5164
	GERNERMED++ @ German BERT	.9805	.9859	.9832	.9737	.9791	.9764
	GERNERMED++ @ GottBERT	.9788	.9787	.9787	.9656	.9655	.9656
	GERNERMED++ @ Spacy Token2Vec	.9639	.9728	.9683	.9444	.9532	.9488
	GPTNERMED @ GermanMedBERT	.6768	.8211	.7420	.6113	.7417	.6702
	GPTNERMED @ GottBERT	.7173	.8457	.7762	.6396	.7542	.6922
	GPTNERMED @ GBERT	.5764	.9092	.7056	.5205	.8210	.6371
	ATC $\omega_{unk} = 0.01$	.8104	.7897	.7999	.6714	.6542	.6627
	ATC $\omega_{unk} = 0.05$	.8363	.7383	.7843	.7540	.6656	.7071
	ATC $\omega_{unk} = 0.1$	.7627	.7654	.7640	.7043	.7068	.7056
	ATC $\omega_{unk} = 0.2$	.8453	.7526	.7963	.7645	.6807	.7201
	ATC $\omega_{unk} = 0.5$	.8518	.7152	.7776	.7846	.6588	.7162
	ATC $\omega_{unk} = 0.8$	.8773	.6876	.7710	.8122	.6367	.7138
	ATC $\omega_{unk} = 1.0$	.8831	.6841	.7710	.8074	.6254	.7049
GPTNERMED [Drug = `Medikation`]							
Models	DrNote	.9569	.7404	.8349	.9367	.7248	.8173
	GERNERMED	.7751	.5669	.6548	.6150	.4498	.5196
	GERNERMED++ @ German BERT	.8636	.8978	.8803	.8235	.8561	.8395
	GERNERMED++ @ GottBERT	.8699	.9017	.8855	.8329	.8633	.8479
	GERNERMED++ @ Spacy Token2Vec	.7979	.7588	.7778	.7709	.7331	.7515
	GPTNERMED @ GermanMedBERT	.9935	.9878	.9906	.9685	.9630	.9657
	GPTNERMED @ GottBERT	.9927	.9874	.9900	.9675	.9624	.9649
	GPTNERMED @ GBERT	.9882	.9913	.9897	.9626	.9656	.9641
	ATC $\omega_{unk} = 0.01$	.8002	.8802	.8383	.6343	.6977	.6645
	ATC $\omega_{unk} = 0.05$	.8286	.8638	.8458	.7758	.8089	.7920
	ATC $\omega_{unk} = 0.1$	.7410	.8787	.8040	.7099	.8418	.7703
	ATC $\omega_{unk} = 0.2$	.8553	.8537	.8545	.7632	.7618	.7625
	ATC $\omega_{unk} = 0.5$	.8300	.8413	.8356	.8024	.8133	.8078
	ATC $\omega_{unk} = 0.8$	.8145	.8151	.8148	.7819	.7824	.7821
	ATC $\omega_{unk} = 1.0$	.8336	.8172	.8253	.8010	.7853	.7931

Table 5.17: Entity-wise NER evaluation on resulting corpora. BratEval is used for the score computation. Prior to the computation, all unrelated labels are removed and all related labels unified into one, common label class, and a span overlap removal is applied.

NER Results		Drug (overlap/lenient)			Drug (exact)		
		Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
GSC EMEA [Drug = CHEM]							
Models	DrNote	.8571	.3830	.5294	.8333	.3723	.5147
	GERNERMED	.3621	.2234	.2763	.2759	.1702	.2105
	GERNERMED++ @ German BERT	.7045	.6596	.6813	.5795	.5426	.5604
	GERNERMED++ @ GottBERT	.6957	.6809	.6882	.5870	.5745	.5806
	GERNERMED++ @ Spacy Token2Vec	.5385	.2979	.3836	.4615	.2553	.3288
	GPTNERMED @ GermanMedBERT	.7053	.7128	.7090	.6105	.6170	.6138
	GPTNERMED @ GottBERT	.7126	.6596	.6851	.6207	.5745	.5967
	GPTNERMED @ GBERT	.6667	.7872	.7220	.5946	.7021	.6439
	ATC $\omega_{unk} = 0.01$	.5370	.6170	.5743	.4352	.5000	.4653
	ATC $\omega_{unk} = 0.05$	.6250	.5851	.6044	.5455	.5106	.5275
	ATC $\omega_{unk} = 0.1$	.5446	.5851	.5641	.4554	.4894	.4718
	ATC $\omega_{unk} = 0.2$	.6310	.5638	.5955	.5476	.4894	.5169
	ATC $\omega_{unk} = 0.5$	.6761	.5106	.5818	.5775	.4362	.4970
	ATC $\omega_{unk} = 0.8$	.6026	.5000	.5465	.5000	.4149	.4535
	ATC $\omega_{unk} = 1.0$	.6522	.4787	.5521	.5797	.4255	.4908
GSC Medline [Drug = CHEM]							
Models	DrNote	.8125	.2600	.3939	.8125	.2600	.3939
	GERNERMED	.2692	.2800	.2745	.1154	.1200	.1176
	GERNERMED++ @ German BERT	.5238	.6600	.5841	.3492	.4400	.3894
	GERNERMED++ @ GottBERT	.5067	.7600	.6080	.3067	.4600	.3680
	GERNERMED++ @ Spacy Token2Vec	.1774	.2200	.1964	.1613	.2000	.1786
	GPTNERMED @ GermanMedBERT	.4717	.5000	.4854	.3774	.4000	.3883
	GPTNERMED @ GottBERT	.7436	.5800	.6517	.6154	.4800	.5393
	GPTNERMED @ GBERT	.5507	.7600	.6387	.3623	.5000	.4202
	ATC $\omega_{unk} = 0.01$	.2672	.6200	.3735	.1552	.3600	.2169
	ATC $\omega_{unk} = 0.05$	.3614	.6000	.4511	.2289	.3800	.2857
	ATC $\omega_{unk} = 0.1$	.2793	.6200	.3851	.1892	.4200	.2609
	ATC $\omega_{unk} = 0.2$	.4844	.6200	.5439	.2813	.3600	.3158
	ATC $\omega_{unk} = 0.5$	.3523	.6200	.4493	.2273	.4000	.2899
	ATC $\omega_{unk} = 0.8$	.2828	.5600	.3758	.1919	.3800	.2550
	ATC $\omega_{unk} = 1.0$	.3836	.5600	.4553	.2740	.4000	.3252
GSC Patent [Drug = CHEM]							
Models	DrNote	.3462	.0625	.1059	.2308	.0417	.0706
	GERNERMED	.1584	.1111	.1306	.0594	.0417	.0490
	GERNERMED++ @ German BERT	.4444	.2778	.3419	.2667	.1667	.2051
	GERNERMED++ @ GottBERT	.5303	.2941	.3784	.3636	.2017	.2595
	GERNERMED++ @ Spacy Token2Vec	.2727	.1042	.1508	.2000	.0764	.1106
	GPTNERMED @ GermanMedBERT	.4730	.2431	.3211	.3378	.1736	.2294
	GPTNERMED @ GottBERT	.5385	.4118	.4667	.3626	.2773	.3143
	GPTNERMED @ GBERT	.4481	.4792	.4631	.3506	.3750	.3624
	ATC $\omega_{unk} = 0.01$	.3564	.2500	.2939	.1683	.1181	.1388
	ATC $\omega_{unk} = 0.05$	.4268	.2431	.3097	.2073	.1181	.1504
	ATC $\omega_{unk} = 0.1$	.5366	.1528	.2378	.3659	.1042	.1622
	ATC $\omega_{unk} = 0.2$	.5652	.1806	.2737	.3261	.1042	.1579
	ATC $\omega_{unk} = 0.5$	.6471	.1528	.2472	.4118	.0972	.1573
	ATC $\omega_{unk} = 0.8$	.6765	.1597	.2584	.4412	.1042	.1685
	ATC $\omega_{unk} = 1.0$	.4848	.1111	.1808	.2727	.0625	.1017

Table 5.18: Entity-wise NER evaluation on the GSC corpora. BratEval is used for the score computation. Prior to the computation, all unrelated labels are removed and all related labels unified into one, common label class, and a span overlap removal is applied.

	Bronco150	CARDIO:DE	GGPOnc 1.0	GGPOnc 2.0 (fine, short)	OoD	GERNERMED++	GPTNERMED	ATC (raw)	GSC EMEA	GSC Medline	GSC Patent
DrNote	.6583	.5968	.6420	.6016	.7879	.7153	.8349	.7210	.5294	.3939	.1059
GERNERMED	.2991	.2759	.1761	.1517	.5333	.7359	.6548	.3170	.2763	.2745	.1306
GERNERMED++ @ German BERT	.7270	.6968	.5989	.4495	.9091	.9832	.8803	.5744	.6813	.5841	.3419
GERNERMED++ @ GottBERT	.7708	.7455	.6093	.4643	.9091	.9787	.8855	.5667	.6882	.6080	.3784
GERNERMED++ @ Spacy Token2Vec	.4441	.3652	.2948	.2349	.7105	.9683	.7778	.3744	.3836	.1964	.1508
GPTNERMED @ GermanMedBERT	.6420	.5609	.6311	.4330	.8395	.7420	.9906	.5436	.7090	.4854	.3211
GPTNERMED @ GottBERT	.7407	.5339	.6183	.4739	.8193	.7762	.9900	.5294	.6851	.6517	.4667
GPTNERMED @ GBERT	.6216	.4064	.5053	.3527	.7692	.7056	.9897	.5061	.7220	.6387	.4631
ATC $\omega_{unk} = 0.01$	.7792	.6197	.6597	.3021	.8732	.7999	.8383	.8673	.5743	.3735	.2939
ATC $\omega_{unk} = 0.05$	.7729	.5895	.6408	.3552	.8611	.7843	.8458	.8937	.6044	.4511	.3097
ATC $\omega_{unk} = 0.1$	.7745	.6447	.6297	.2935	.8533	.7640	.8040	.8903	.5641	.3851	.2378
ATC $\omega_{unk} = 0.2$	.7768	.6106	.5498	.3442	.8732	.7963	.8545	.9110	.5955	.5439	.2737
ATC $\omega_{unk} = 0.5$	.7464	.6163	.4686	.3128	.8611	.7776	.8356	.9233	.5818	.4493	.2472
ATC $\omega_{unk} = 0.8$	.6980	.5985	.3884	.3131	.8732	.7710	.8148	.9304	.5465	.3758	.2584
ATC $\omega_{unk} = 1.0$	.7035	.5775	.3974	.3402	.8116	.7710	.8253	.9338	.5521	.4553	.1808

Table 5.19: Lenient entity-wise F1-scores. Each row displays the scores for a single model on individual corpora as ground truth (columns). BratEval is used for the score computation. Prior to the computation, all unrelated labels are removed and all related labels unified into one, common label class, and a span overlap removal is applied.

	Bronco150	CARDIO:DE	GGPOnc 1.0	GGPOnc 2.0 (fine, short)	OoD	GERNERMED++	GPTNERMED	ATC (raw)	GSC EMEA	GSC Medline	GSC Patent
DrNote	.5670	.5785	.6256	.5577	.6970	.6072	.8173	.7078	.5147	.3939	.0706
GERNERMED	.2299	.2467	.0957	.0862	.3733	.5164	.5196	.1459	.2105	.1176	.0490
GERNERMED++ @ German BERT	.6611	.6708	.5250	.4132	.8312	.9764	.8395	.4672	.5604	.3894	.2051
GERNERMED++ @ GottBERT	.7098	.7146	.5362	.4285	.8052	.9656	.8479	.4810	.5806	.3680	.2595
GERNERMED++ @ Spacy Token2Vec	.4088	.3597	.2528	.2237	.5789	.9488	.7515	.3217	.3288	.1786	.1106
GPTNERMED @ GermanMedBERT	.5931	.5591	.5614	.4164	.7160	.6702	.9657	.4981	.6138	.3883	.2294
GPTNERMED @ GottBERT	.6888	.5243	.5283	.4421	.7229	.6922	.9649	.4824	.5967	.5393	.3143
GPTNERMED @ GBERT	.5764	.4007	.4341	.3309	.6374	.6371	.9641	.4587	.6439	.4202	.3624
ATC $\omega_{unk} = 0.01$	.5934	.5707	.5585	.2650	.7042	.6627	.6645	.7308	.4653	.2169	.1388
ATC $\omega_{unk} = 0.05$	.6402	.5556	.5631	.3346	.7222	.7071	.7920	.7948	.5275	.2857	.1504
ATC $\omega_{unk} = 0.1$	.7083	.6324	.5653	.2789	.7200	.7056	.7703	.8049	.4718	.2609	.1622
ATC $\omega_{unk} = 0.2$	.7041	.5908	.4919	.3209	.7042	.7201	.7625	.8244	.5169	.3158	.1579
ATC $\omega_{unk} = 0.5$	.6949	.6134	.4234	.2991	.7222	.7162	.8078	.8403	.4970	.2899	.1573
ATC $\omega_{unk} = 0.8$	.6444	.5925	.3601	.3015	.7042	.7138	.7821	.8478	.4535	.2550	.1685
ATC $\omega_{unk} = 1.0$	.6498	.5730	.3695	.3293	.6667	.7049	.7931	.8506	.4908	.3252	.1017

Table 5.20: Exact entity-wise F1-scores. Each row displays the scores for a single model on individual corpora as ground truth (columns). BratEval is used for the score computation. Prior to the computation, all unrelated labels are removed and all related labels unified into one, common label class, and a span overlap removal is applied.

5.6 Open-source Repositories and Web Resources

The presented methods and results, including the obtained resources, have been developed within the context of this thesis as original work. The publicly accessible data, models, and code of this work are referenced in this section.

For the work centered around the search-based approach, DrNote, the required code that has been used to initialize and start the web-based annotation service is available as a code repository on GitHub[10]. To improve accessibility, a technical demonstration web page has been developed, accompanying the GitHub repository and enabling interactive testing and exploration[11]. A screenshot of the web page is shown in Figure 5.15.

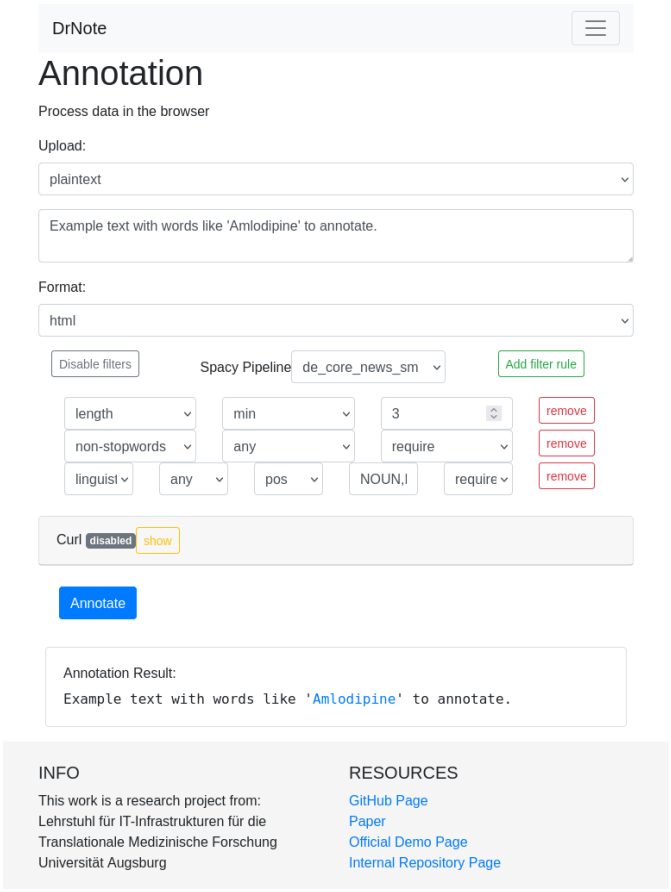

Figure 5.15: Interactive demonstration page for the DrNote implementation.

The model weights, initial resources, and demo code samples for both translation- and word-alignment-based approaches, GERNERMED and GERNERMED++, are provided at their respective GitHub repositories[12,13]. The OoD dataset is provided at the latter repository. Similarly, a web demo has been developed to provide immediate and

[10] https://github.com/frankkramer-lab/DrNote (accessed May 9th, 2025)
[11] https://drnote.misit-augsburg.de/ (accessed May 9th, 2025)
[12] https://github.com/frankkramer-lab/GERNERMED (accessed May 9th, 2025)
[13] https://github.com/frankkramer-lab/GERNERMED-pp (accessed May 9th, 2025)

interactive access[14].

Similar to the public assets for the translation- and word-alignment-based approaches, the approach utilizing an LLM-generated corpus is accompanied by a GitHub repository[15], as well as an interactive demonstration page[16] as shown in Figure 5.17.

The former four repositories provide the model data as well as associated information and resources. Yet the majority of the developed source code of the original work has not been publicly shared at the time of writing, including the data preprocessing pipelines, model training code, and evaluation pipelines. Because the model weights are publicly shared, the foundational elements needed to reproduce the quantitative evaluation results are available.

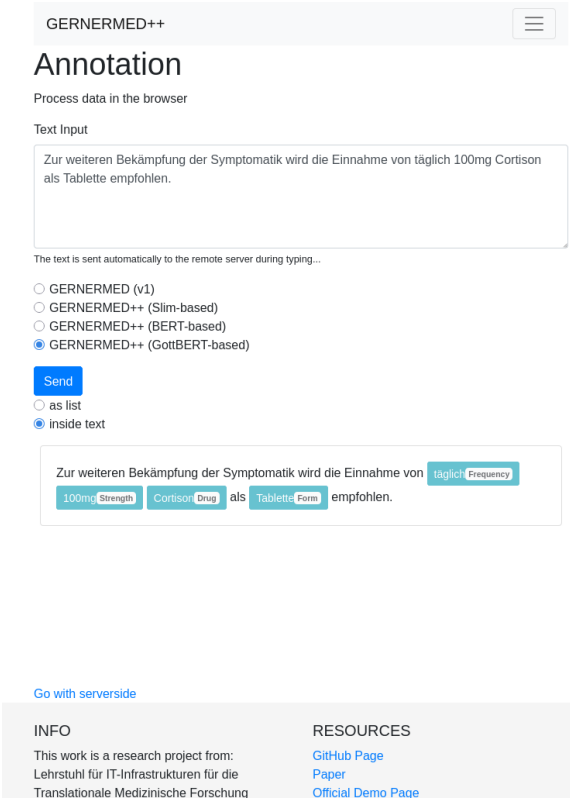

Figure 5.16: Interactive demonstration page for the GERNERMED and GERNERMED++ models.

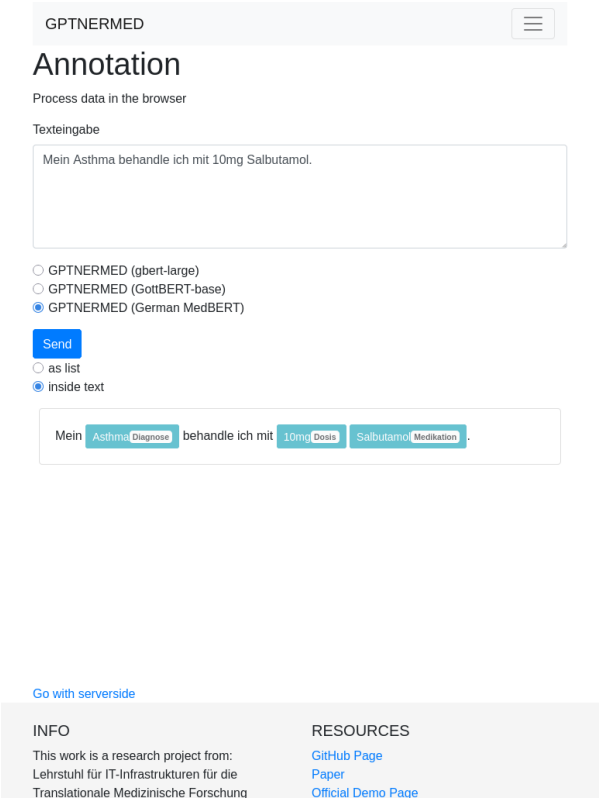

Figure 5.17: Interactive demonstration page for the GPTNERMED models.

The Wikipedia-based approach, which uses the weakly-annotated Wikipedia and WikiData resources, is fully publicly available as source code at the corresponding GitHub repository[17], covering the code for corpus synthesis, NER training code and model evaluation. While the text corpora are published, the model weights of the fine-tuned NER models are not shared due to unclear licensing surrounding the *medBERT.de* model from GerMedBERT.

[14] https://gernermedplusplus.misit-augsburg.de/ (accessed May 9th, 2025)
[15] https://github.com/frankkramer-lab/GPTNERMED (accessed May 9th, 2025)
[16] https://gptnermed.misit-augsburg.de/ (accessed May 9th, 2025)
[17] https://github.com/frankkramer-lab/WikiOntoNERCorpus (accessed May 9th, 2025)

To lower the barrier for other researchers to utilize the annotated corpus synthesis from Wikipedia texts, a web application has been developed to run this synthesis step for external users, supporting English, German, French, and Spanish Wikipedia corpora[18]. The application provides a text input for a valid, user-defined SPARQL query, and subsequently provides the corresponding, annotated dataset to the user within certain runtime limits (<1h processing time per task). The web application is shown in Figure 5.18.

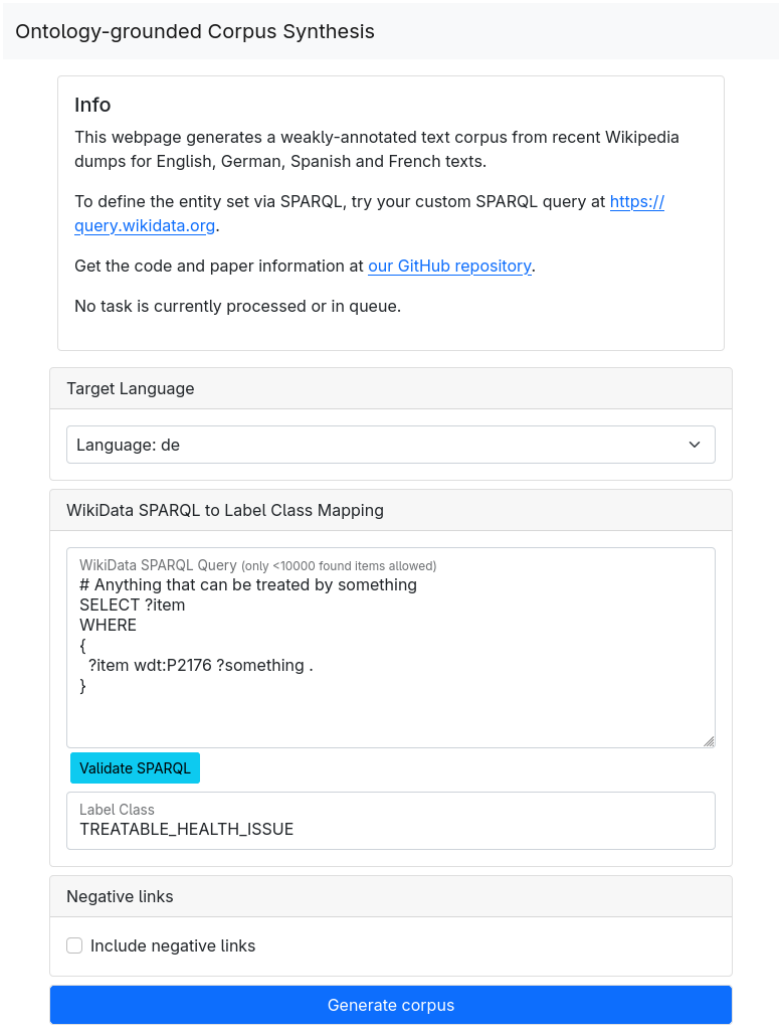

Figure 5.18: Interactive web application for generating custom SPARQL-defined, Wikipedia-based annotated corpora. The page supports English, German, French, and Spanish Wikipedia dumps.

The web applications have been primarily developed using React[19] and Flask[20].

[18] https://ontocorpus.misit-augsburg.de/ (accessed May 9th, 2025)
[19] https://react.dev/ (accessed May 9th, 2025)
[20] https://flask.palletsprojects.com/ (accessed May 9th, 2025)

Chapter 6

Discussion

Returning to the key motivations of the original research objective, several factors are critical to the design of the methods developed. To this end, it involves the following points. First, the focus on German texts poses an explicit challenge, as the majority of research attention in NLP is naturally drawn to English and other non-German languages and relevant materials and datasets are comparatively scarce for German, emphasizing the issue of limited language resources. Second, the rather efficient use of hardware resources to enable low-latency, high-throughput model inference in practical applications is a central objective of this thesis. It highlights a clear contrast to current trends of deploying compute- and memory-intensive LLMs, which follow the autoregressive language modeling design and are often applied by framing traditional NLP tasks as question-answering problems. Since local deployment of NER models can be mandatory for the processing of patient-related data, models that exceed certain hardware limits may become impractical. Third, all proposed methods are intended to avoid conceptual steps that immediately involve substantial amounts of additional manual effort. For instance, this includes manual data annotation as a common procedure in NLP to obtain training data for supervised learning techniques. Fourth, the exclusion of internal and proprietary data or highly access-restricted data sources is a highly crucial aspect, in particular in the medical domain where openly and publicly available data resources tend to be scarce and the utilization of internal hospital data remains challenging. Patient-related hospital data is subject to strict restrictions on data retrieval, processing and sharing, and subsequently inflicts further issues such as reproducibility concerns from a scientific perspective.

Concerning the stated goals, all four proposed approaches satisfy these aspects largely. However, they differ fundamentally in their implementations and to some extent in their achieved NER performance. Therefore, the findings from the conducted experiments are discussed within the narrow focus of the medication detection benchmark as well as in a broader context in this chapter.

6.1 Principal Findings

The general findings drawn from the results, partly mentioned in the previous chapter, are as follows.

- **Acceptable to Remarkable Performance**: Nearly all proposed methods can generally be considered to perform well, given the complexity and ambiguity of the medication detection task. For the Bronco150 corpus, the best models from the various neural approaches exceed the .70 lenient F1-score threshold, and the search-based baseline approach reaches around .65 in lenient F1-score performance. For the small OoD benchmark, the obtained results can even further exceed these scores and partially reach scores well above the .80 lenient F1-score threshold.

- **Performance Hierarchy Trends**: When comparing the different methods based on their best lenient F1-score performance, the translation- and word-alignment-based approach shows the most promising results. These are closely followed by the weakly-annotated, Wikipedia-sourced method, then by the method relying on LLM-synthesized training data. The search-based method, used as the baseline approach, remains the least well-performing method. However, while these findings can be drawn from the obtained F1-score results, it needs to be emphasized that a multitude of factors impact the final F1-scores, and changes with regard to these factors could lead to different conclusions.

- **Superiority of Transformer-based Models**: The findings of the results clearly confirm the common theme in the NLP research about the strong performance of the models that rely on BERT encoders compared to more traditional, non-transformer-based neural architectures like the Token2Vec encoders. In particular, both encoder types seem to fit the training data sufficiently well, yet evidently the Token2Vec encoders struggle notably to generalize to other text corpora compared to the used BERT encoders.
  The question of which BERT model performs best among all German monolingual BERT models remains opaque since different BERT models are used on different datasets due to their varying availability during the development of the methods. Yet the findings highlight that large-sized BERT models do not necessarily yield superior NER performance over smaller-sized BERT models. In addition, domain-specific models such as German MedBERT may contribute to a better NER performance on some corpora but do not universally improve the final results. While German MedBERT relies on the German model and essentially adds another domain-specific pre-training stage, it does not take potential advantage of using a domain-optimized subword tokenizer. However, the reports from the medBERT.de model suggest that an improved tokenizer fertility does not necessarily correlate with enhanced performance [63].

6.2 Method-specific Discussion

6.2.1 Search-based Approach

- **Performance Benefits**: Although most BERT models are substantially smaller and more practical to deploy on low-resource hardware when compared to most LLMs, even small BERT models remain computationally demanding in contrast to simple rule-based or smaller neural systems. Therefore, the Solr-powered tagger engine combined with the SVM of OpenTapioca and the generic text parsing pipeline from Spacy can be implemented to run even with very limited computational resources. Crucially, this affects latency and throughput measures and thus might be preferred in certain contexts where the annotation quality is not necessarily the primary objective.
- **Entity Linking**: OpenTapioca encompasses an EL system rather than a pure NER system. Therefore, the measured NER results are drawn from the EL functionality. This largely ignores the potential advantage that EL systems provide over NER systems. To transform the predicted EL annotations into NER spans, the EL references are assigned with a class label if their linked WikiData item is part of a predefined set of WikiData entities that defines a certain label class. Changing such a SPARQL-based definition immediately influences the NER annotation results and thus its obtained evaluation scores.
- **Ongoing WikiData Editing**: In WikiData, new entities and relations are constantly added or updated over time. This could also entail severe changes to the graph structure and can lead to completely different results to the identical SPARQL query at different points in time. Therefore, the obtained results, including the F1-scores, can also vary depending on the applied SPARQL query and the current state of the WikiData graph. Due to the large size of the graph and its implications on the practicability of deploying a fixed WikiData snapshot, using a fixed snapshot locally was abandoned during the experiments and the public SPARQL web endpoint was preferred in the context of this work.
- **Knowledge Base Quality**: Relying on WikiData as a knowledge base also has broader consequences with respect to the quality of the final NER pipeline. While WikiData features a open and easily accessible foundation for universal knowledge representation, more domain-specific, high-quality ontologies may be better suited for tasks like medication detection. However, beyond its openness and universality, other characteristics may also be preferable. For instance, WikiData also covers common but informal concept labels that may not be fully captured by domain-specific terminology systems.
- **Pre- and Postprocessing**: A generic text parsing pipeline for German text is used as a postprocessing step in order to filter out false positive EL results based on a basic set of linguistic assumptions. In this regard, more sophisticated processing steps may be able to further improve upon the pre-existing results.

- **SVM-based Training**: The final outcome of the EL pipeline relies on the SVM-based classifier to rank and potentially reject linking candidates. In order to learn a suitable decision boundaries, OpenTapioca requires a training corpus that is fully annotated with WikiData entities, yet in the experiments, only a weakly annotated corpus is used. Thus, the training may not yield as stable and robust results as it could on manually annotated texts.

6.2.2 Translation-based Approach

- **Limited Corpus Availability and Flexibility**: The results indicate that an approach relying on NMT and word alignment can provide the best results of all proposed methods. Contrary to other methods however, the usefulness of this approach depends on the availability of an annotated corpus in a foreign language with relevant annotation classes that fit the target use case. Furthermore, certain datasets may be publicly available for research purposes but its use for commercial applications may be prohibited.

- **Translation Quality**: The quality of the available NMT system may heavily affect the outcome of the approach. While English to German NMT is a rather well-supported task in current NMT systems, more low-resource languages with regard to both the source and the target language may worsen the results. Similarly, the word alignment strongly affects the quality of the projected NER annotations. These word alignment models tend to perform well in rather high-resource contexts such as for English-German sentence pairs but may deteriorate in contexts with low-resource languages.

- **Implementation Refinements**: The improvements in the implementation details have a substantial impact on the final NER performance outcome of the resulting models. In particular, through the use of both the BERT-based encoders for transfer learning and the improved word tokenization and word alignment method combined, the second iteration approach is shown to be superior to the first iteration approach.

6.2.3 LLM-synthesized Corpus-based Approach

- **LLM Quality**: As the research on LLMs has progressed tremendously in recent years, a number of novel open LLMs have been published that have been shown to further improve common LLM capabilities on a set of NLP benchmarks, including text generation. The quality of the synthesized corpus, which includes both the text and the accompanying NER annotations, can strongly affect the NER performance of models fine-tuned on these data samples. Given that the GPT NeoX 20B model has been surpassed by more recent, more compact LLMs,

including Meta's Llama 3.1, on numerous benchmarks[1], the potential impact of these more advanced LLM models on dataset synthesis is likely to result in notable overall quality enhancements.

- **Lack of Structured Decoding**: The current implementation for synthesizing the text samples with additional NER annotations relies on the encoding of the plain text in accordance with the grammar of the basic hypertext markup. Because deviations from the grammar structure effectively cease the ability to parse the output text and extract the annotation metadata, it is essential to identify a temperature τ parameter that reduces the probability of predicting a token sequence that breaks the markup syntax, while at the same time ensures the generation of diverse text samples that best represent the entirety of the implicitly targeted dataset manifold. By adopting an implementation of a structured decoding mechanism, the remaining sampling parameters such as the temperature τ could be chosen with an increased focus on text diversity. The current results indicate that roughly 66% of the removed sample lines were due to duplicated lines, indicating that the chosen sampling parameters were selected rather conservatively. However, the fact that less than one percent of removed sample lines were due to invalid syntax does not imply that syntactical breaks occurred very infrequently. It is plausible that lines with severe syntax flaws could also lack the closing tag `</s>` and therefore were already removed in the first filtering stage, accounting for 15% of all removed sample lines.

- **Promising Flexibility**: In contrast to the other proposed methods, the LLM-generated corpus is clearly superior in its flexibility to cover arbitrary label classes as it is not tied to a predefined concept structure or to pre-annotated label class definitions of an annotated corpus in a foreign language. The utilisation of the LLM as a data augmentation method does not suffer from the same limitations in terms of flexibility as other approaches, and thus could be employed in a more universally adaptable manner.

- **Markup Optimization**: Currently, the proposed hypertext markup implements a very basic syntactic structure to encode annotation-related entities. Other approaches to encode metadata information are thus underexplored and might be beneficial for improving the annotation of named entities, as well as for other tasks such as the encoding of relation extraction information, which is not addressed by the current markup syntax.

- **Instruction-based LLMs and Prompt-Tuning**: The utilization of LLMs that are further fine-tuned on instruction-oriented prompt templates is not regarded by the current method. Instead, the GPT NeoX 20B model rather represents a raw LM without instruction-based fine-tuning. Rather than relying solely on

[1]GPT NeoX 20B reaches an averaged score of 5.99 points whereas Meta's Llama 3.1 8B reaches an averaged score of 13.78 points according to the Open LLM Leaderboard page (https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard) (accessed October 25th, 2024). The averaged score encompasses multiple benchmark scores.

implicitly expressed label class definitions and annotation rules through pure text continuation in the current approach, adding relevant annotation guideline information to the prompt may help to ensure a more fine-grained and controlled annotation style over the current approach. Even without adding an explicit annotation guideline to the prompt, only a re-phrasing of the label class names has been shown to have an impact on the LLM's behavior and should be further investigated in future work [142].

- **Robustness Considerations**: A substantial limitation remains the implicit, example-driven label class definition that also complicates the distinction of edge cases that are or are not supposed to be part of a certain label class. The LLM-based data augmentation may penalize borderline entity samples over the mention of more commonly and ordinary concepts due to its probability-based token sampling (e.g. considering P("`cortisone`") $\gg$ P("`thiopentone`") for illustration purposes), and hence, less-known terms and edge-cases may be underrepresented and further blur the training data-provided evidence for an NER model in borderline cases. In this context, it is worth emphasizing that label classes can be affected to different extents, and plausibly the definition of less complex label classes as well as distinct label class names eventually also contribute to an increased NER performance.
- **Hallucination Issues**: In the context of LLMs, the ongoing issue of hallucinations in principle applies to the proposed approach as well. This issue can surface as plainly wrong annotations as well as obscure or even nonsensical text. To what extent potential LLM-inflicted hallucinations impede the resulting dataset and the trained NER models have not been investigated systematically in this work and could be further addressed in future work.

6.2.4 Wikipedia-sourced Weakly-annotated Approach

- **Intervention-less, Ontology-defined NER**: The proposed method is implemented as a fully automated pipeline and does not require any manual intervention. Ultimately, it outputs a fine-tuned NER model based on one or multiple user-defined SPARQL queries per label class, allowing customized entity extraction based on specific ontological patterns. This concept enables an adaptable model custom-fine-tuned to any SPARQL-defined entity sets present in WikiData, leveraging the structured information of its graph structure, and offers a well-defined and technical alternative to annotation guidelines that can suffer from informal label definitions.
- **Use of Wikipedia and WikiData**: By utilizing open data sources such as Wikipedia and WikiData, the approach can continually benefit from future expansions and updates to these resources. The method leverages WikiData's structured ontology jointly with Wikipedia's textual content, allowing for the extraction of rich, entity-linked training data and, due to its fully automated pipeline, the

method can adapt to evolving content with minimal maintenance. To this end, a major advantage of these data sources is their openness and their public domain licensing status which, contrary to typical clinical corpora, facilitates a direct data and model sharing as well as it eases the reproducibility aspect.

- **Limited Flexibility**: The proposed method is inherently restricted to entities defined within the WikiData ontology and Wikipedia pages, which may limit its applicability to factual concepts and named entities. While certain entity sets may be well represented within WikiData's graph structure, other label classes may not be covered well by its structure. For example, potential label classes like `Route` or `Frequency` which are present in the n2c2 2018 ADE corpus are difficult to capture using the proposed approach.

- **Stylistic Bias in Text and Annotation**: Wikipedia's editorial style, including its formal language and neutral tone, tends to represent a relatively homogeneous text style. Similarly, Wikipedia's annotation style may differ from common annotation styles in NLP. This can lead to NER models that may not align well with informal or domain-specific corpora, such as clinical text, where narrative style and terminology are unique, and abbreviations are common. Consequently, the approach may require additional fine-tuning to generalize well to non-neutral, domain-specific styles.

- **Remaining Robustness and Hyperparameter Issues**: The results indicate varying robustness of the fine-tuned models on different corpora, which can partly be attributed to poor performance on samples with no annotated entities or very sparsely annotated entities. To a certain extent, this robustness issue can be alleviated by incorporating multiple label classes into the corpus in order to accustom an NER model to samples without the presence of a particular label class [143]. With regard to the τ_{unk} hyperparameter, the positive effect of the token-wise loss scaling is clearly observable in the final F1-scores. However, no clear recommendation can be directly concluded for an optimal choice of this hyperparameter. Only a small set of training configurations was explored in this work, and no repeated training sessions, for instance through an n-fold cross-validation, were further conducted. As a result, this aspect remains relevant for future work. This includes also the investigation of the behavior for fringe hyperparameter choices. For instance, selecting $\tau_{unk} = 0.0$ effectively collapses the training into a contrastive learning task with learning from only positive and negative text spans, excluding all unknown token positions.

- **Differences in Medication-related Entity Definition**: Both the search-based baseline approach as well as the token-wise loss-scaling based approach use WikiData and SPARQL to define the entity set intended to represent all medication-related entities. However, both approaches were developed using different SPARQL queries and at different points in time. This limits the direct comparability of both approaches as they utilize different entity sets as their definition of

medication-related entities. In the token-wise loss-scaling approach, a very strict definition is chosen, as only entities with an assigned ATC code are classified as medication entities. A further optimization of the SPARQL-based definition could further improve the results, but this is not addressed in this work and is considered future work.

- **Language Resource Dependency**: The approach's effectiveness may vary with the availability and quality of language-specific Wikipedia resources. High-resource languages with extensive Wikipedia pages yield better performance, whereas low-resource languages with limited Wikipedia coverage may lead to lower accuracy, reducing the method's generalizability across different linguistic contexts.

6.3 General Discussion and Limitations

- **LLM-based NER**: In recent years, there has been a growing adoption of LLMs especially due to the advent of strong publicly available LLM models and their ease of use through their prompt-based conditioning mechanism. In this work, no LLM is included as a pure NER annotator model and could be considered a limitation of this work. However, this use of LLMs comes in conflict with the premise of this work to focus on relatively practical and efficient approaches which require modest hardware resources at inference time. Moreover, the concept of an LLM-based NER gives rise to a number of methodological questions regarding the optimal implementation of an NER system using a GPT-like (or T5-like) autoregressive model. This is because the BERT architecture may be regarded as a more natural fit for addressing NER tasks, given that NER is commonly modeled as a token classification task. In contrast, the GPT architecture may not be as well-suited for this purpose. Despite these concerns, the effectiveness of LLMs for NER tasks has been investigated for the medical domain, yet mixed results are reported [144].
- **Multilingual NER**: Multilingual BERT models like the XLM model family or mBERT are found to be able to learn useful multilingual representations which can be further utilized to perform cross-lingual transfer. In practice, a multilingually pre-trained BERT model with an NER head can be fine-tuned on an annotated corpus from a foreign source language and be applied directly to text from the target language without the need for further NER fine-tuning on the target language. The success of this concept depends on the quality of the multilingual model to capture semantic embeddings that can abstract from language-specific semantics of individual words and phrases. A major advantage remains the practical simplicity because the implementation resembles a generic NER fine-tuning process. Therefore, it usually does not involve any custom, German-specific implementation steps, nor does it acquire novel textual resources during the process. Such text resources, like a translated or synthesized corpus, are

beneficial for a more comprehensive understanding of the final NER behavior and for the potential repurposing of the corpus for other contexts. Additionally, multilingual models tend to be larger in their number of trainable parameters in comparison to monolingual models, affecting the required amount of time and compute during training and inference. Nevertheless, multilingual models have been trained in the cross-lingual setting for clinical NER, including German texts, and can be a simple yet effective technique to obtain NER models that achieve NER scores similar to models trained in the traditional, monolingual setting [145]. However, the effectiveness of the cross-lingual transfer may also depend on both the actual source and the target language.

- **Cross-validation and Hyperparameter Search**: Throughout the experiments, primarily well-established default hyperparameters were used, many of which are provided by Huggingface Transformers and Spacy. The identification of optimal parameters by performing an exhaustive hyperparameter search was omitted due to varying hardware availability and pragmatic comparability considerations. Thus, most values were either chosen based on default parameters, or taken from prior or preliminary experiments. While finding optimal configuration values for NER fine-tuning could further improve the outcome of certain experiments, exhaustive hyperparameter searches are not considered the focus of this work and remain future work. In a similar context, another limitation worth mentioning includes the decision to skip a time- and compute-intensive n-fold cross-validation for the fine-tuning of the NER models, which could further improve the reliability of the measured performance scores.

- **OoD Corpus Bias**: Due to the lack of out-of-distribution corpora accompanied by adequate annotation data for an extensive evaluation, the OoD is designed and introduced as evaluation material within this work. This is especially useful to evaluate dataset-specific label classes that are unique to a certain corpus. However, as a consequence the OoD corpus was annotated according to the annotation guidelines of the n2c2 2018 ADE corpus, and is therefore substantially biased towards this corpus.

- **Need for Novel Datasets**: The issue of the availability, sparsity, and suitability of annotated corpora has been highlighted on numerous occasions within this work, both explicitly and implicitly. However, it remains a crucial topic that requires continued emphasis, as it plays a fundamental role in the field of medical NLP in the German language. The absence of such corpora not only impede the development of NLP tools in their role as training resource, but these issues are also reflected when it comes to the role of such text resources as evaluation material. Ultimately, the availability of the external corpora limits the extent to which a developed and custom-tailored NLP model can be independently evaluated.

- **Varying Annotation Guidelines and Styles**: At least at first glance, corpora with

similarly named label classes may appear to be comparable or even interchangeable. However, these annotations are subject to potentially diverging annotation guidelines, which could manifest in low inter-annotator agreements and cause models trained on different corpora to exhibit divergent behavior due to underlying inconsistencies. The development of publicly available corpora is a significant achievement that merits appreciation, and these corpora represent an invaluable resource for evaluation. However, the discrepancies in text styles and the varying annotation guidelines pose a challenge to conducting a fair and reliable evaluation. Investigating the performance scores of a particular NER model on individual, external datasets in isolation limits the interpretability of the results, especially in cases of poor performance scores, because both individual failures of an NER model as well as inconsistencies or divergences in the evaluation corpus may be contributing factors to a degraded score. While certainly a broader variety of annotated text corpora may help to alleviate some interpretability issues in certain contexts, non-congruent annotation designs of external ground truth datasets and individual NER models remain a fundamental limitation with respect to a reliable NER evaluation if no further efforts towards annotation harmonization are invested. It is important to acknowledge that discrepancies and disagreements can naturally arise during the manual annotation of unlabeled corpora. These discrepancies may stem from differing interpretations of complex annotation rules, potentially resulting in the rejection of entire label classes [33].

- **Medication Detection**: Medication detection as an NER task fits well as a demonstration use case especially due to the fact that several corpora are available with label classes semantically *related* to medication entities, but other potentially useful label classes were not further investigated across all experiments as well. In future work, an increased focus on the development and evaluation for other relevant label classes in medical contexts could outline appropriate follow-up steps on this matter.

- **NER-centric Experiments**: The scientific field of NLP includes a vast array of different text processing tasks, and undoubtedly NER only exemplifies one minor subfield of NLP. Including ontologies or terminologies that have been established as medical knowledge bases in the community, such as UMLS or SNOMED, via EL often entails the development of an NER component as an initial step, yet requires additional engineering efforts to implement successfully. In this work, mostly NER methodologies are addressed, and the scope of this work does not cover other, potentially highly relevant tasks for medical NLP that may build upon NER. For instance, tasks like EL, relation extraction, or negation detection could be considered domain-specific yet use case-agnostic NLP tasks that may be required in order to process medical documents. In this regard, cTAKES serves as a notable example of a medical NLP toolbox that can integrate various pre-trained NLP models for common text processing tasks with its primary focus on English data. The development of such a toolbox for German medical texts would not

only require a suite of NER models but also support for other common tasks. Therefore, further research on objectives similar to this work could be extended to a wider range of medical NLP tasks as well.

- **NER Complexity**: The proposed approaches only conform to basic NER scenarios that make strong assumptions about the NER annotation which may not apply in more complex situations. In real-world applications, entities may not always appear as distinct, non-overlapping sequences within the text. More complex scenarios include disjointed entities where an entity spans non-contiguous text segments, nested entities where one entity is embedded within another, and overlapping entities where entities share part of their text span. Addressing these complexities requires more sophisticated model architectures and annotation strategies that can accurately represent these structures, meaning that changes to both the NER architecture and the sourcing of potential training data to harvest more complex types of annotation are required as well. Future work could explore more advanced NER models capable of handling such complex cases to improve the usefulness of medical NER systems in practice.

- **Consistent Evaluation**: Because a fundamental aspect of the quantitative evaluation relies on the evaluation scores, such as precision, recall and F1-scores, the conducted experiments show that fluctuations in these scores are to be expected when subtle changes are made to the design decisions within the evaluation pipeline. In this work, the different evaluation strategies for individual proposed methods stem from the fact that the methods were originally developed independently within isolated research articles. To address this limitation in this work, a unified evaluation section is provided to ensure a consistent and homogeneous quantitative comparison across all proposed methods. However, comparing the score-wise results reliably with other work remains challenging due to the numerous subtle factors that may influence the final scores.
  This challenge arises not only from variations in NER-specific evaluation schemes and strategies but also from preprocessing and postprocessing steps that can impact final NER scores. Such steps may be rooted not only in methodological considerations but also justified by practical engineering constraints. For example, long text documents that exceed common BERT context window sizes (e.g. >512 tokens) cannot be processed directly and therefore require an appropriate text splitting strategy.

- **F1-score-based Evaluation**: The F1-score-centric evaluation framework is a well-established standard for the scientific evaluation of NER models within NLP. However, in practical applications, final model selection often considers factors beyond F1-score. These include hardware requirements, processing latency, throughput efficiency, and qualitative aspects such as the method's behavioral predictability and explainability, which may be critical depending on the use case.

In conclusion, while several significant elements and limitations have been actively

highlighted, providing a comprehensive overview of all potential discussion points remains unfeasible due to the multitude of factors influencing the final outcome.

Chapter 7

Conclusion

Given the constraints introduced as a premise of this work, four different approaches were proposed to develop NER models for medical texts in German. All approaches fulfill the stated requirements that the methodology (i) should not rely on internal, proprietary data, (ii) avoids costly efforts of manual text annotation, and (iii) can be implemented and applied without substantial hardware resources. As part of the research objective, the developed approaches were evaluated independently, as well as comparatively on several datasets, including external corpora, on the medication detection task due to its suitability given the availability of appropriate annotation data.

The development of the four proposed methods constitutes a major part of the contributions, and their core concepts can be outlined as follows.

- **Search-based Entity Linking**: The WikiData knowledge base is leveraged to detect medical entities using a text search approach.
- **Translation and Annotation Projection**: An annotated English medical corpus is translated into German, while the annotation metadata is re-used and projected onto the German sentences via word alignment. The resulting data is used to fine-tune an NER model.
- **LLM-based Data Synthesis**: An LLM is used for data augmentation to generate annotated text samples that serve as training data to be used to fine-tune an NER model.
- **Wikipedia-sourced Data Synthesis**: The hyperlinks from the Wikipedia texts are re-purposed as annotation information and extracted as weakly-annotated training data. A custom-designed NER training on the data yields a fully-annotated training corpus which is used to fine-tune an NER model.

Together, these methods provide a diverse set of strategies that address the critical bottleneck of data scarcity in German medical NLP and offer practical solutions that can be adapted to similar low-resource settings.

Summarizing the results and contributions, all approaches can achieve competitive results but the findings also indicate a general downward trend in F1-score performance.

The most reliable method tends to be the translation and word-alignment-based approach, followed by the neural Wikipedia-sourced NER model, the NER model trained on LLM-generated data, and finally by the search-based baseline model. However, as the potential downstream tasks are numerous and varied, it is not possible to provide a single, universal recommendation based solely on these results. The proposed methods also exhibit different degrees of flexibility with respect to their applicability, which further complicates the drawing of a general recommendation. In addition, it should be considered that a well-performing F1-score may not be the only objective when selecting a suitable NER method, and other considerations such as compute latency and throughput may remain relevant factors as well.

Other principal findings confirm the prevailing trend in literature and NLP research in general to focus mostly on transformer-based models. When comparing basic neural models with BERT-based models for encoding tokens into semantic embeddings, the results clearly demonstrate the superiority of BERT-based encoders. Nevertheless, basic neural token encoders remain relevant due to their computational efficiency in NER tasks with lower semantic complexity, where they achieve results comparable to those of BERT-based encoders.

The notorious sparsity of available data in the niche domain of German medical NLP limits both the development of NLP models and a further, more comprehensive evaluation of the developed tools and models, effectively rendering the data aspect the most critical issue in this applied field. Since it is crucial to independently evaluate the performance of trained models, which includes the need to track their generalizability to texts with other styles and linguistic properties, a continued pursuit of the collection of more diverse and multi-sourced evaluation data remains a vital and ongoing objective for the research community in this field, with the ultimate goal of establishing a common NLP benchmark.

Although the need for more diverse evaluation corpora is emphasized in this work, the problematic aspect of diverging annotation labels in the context of medical NER is discussed in this work as well. In particular, label classes across multiple evaluation corpora may appear semantically related but may diverge subtly or even substantially according to their respective annotation guidelines, and thus can limit the reliability of the F1-score performance on the evaluation corpora. Additionally, certain label classes may not be present in certain evaluation corpora. This rules out an extensive evaluation of a developed model on such label classes. Because new label classes can be defined arbitrarily, this problem cannot be addressed universally. Yet the issue of varying label classes with regard to medication detection, as well as the issue of missing corresponding label classes in the evaluation data both remain a limitation in the context of this work. As future work, an increased focus on standardized annotation guidelines and label classes may partially solve the stated issues, however it requires the community-wide acceptance and adoption of these standards.

Regarding future developments in the NLP domain, it remains an open question

whether LLM-based, prompt-controlled techniques will outpace more traditional supervised approaches like BERT, as they do not necessarily require additional training data or fine-tuning to be applied, and are easily accessible without the need for deeper NLP knowledge. However, in terms of compute latency, throughput, and compute efficiency, which ultimately translate into energy costs, smaller, use-case-specific models like BERT can remain advantageous in certain scenarios.

Providing methods and tools to address the low-resource context in practical NLP applications is a valuable contribution to the current landscape. By focusing on publicly available resources, the proposed methods can be re-used and replicated for real-world applications, and novel models could potentially be shared within the research community. It can further improve the availability of tools and models which can be leveraged by researchers and other users. The focus on relatively small NLP models, such as BERT-based transformers for NER, also enables greater resource efficiency, making them particularly well-suited for use cases like data integration where high throughput and computational efficiency are critical.

In conclusion, this thesis addresses a critical gap in the development of NER models for German medical texts and presents a set of novel methodologies tailored to the challenges of low-resource NLP. By combining resource-efficient techniques with creative solutions to data scarcity, this work advances the current state of medical NLP for German and lays the groundwork for further exploration and application. Ultimately, these contributions represent a meaningful step toward the broader goal of enhancing healthcare information systems, improving access to medical knowledge, and supporting better patient outcomes in German-speaking regions and beyond.

Bibliography

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, ISSN: 1476-4687. DOI: 10.1038/nature14539.

[2] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 17, 2015. DOI: 10.1126/science.aaa8415.

[3] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, "A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955," *AI Magazine*, vol. 27, no. 4, pp. 12–12, Dec. 15, 2006, ISSN: 2371-9621. DOI: 10.1609/aimag.v27i4.1904.

[4] S. J. Russell and P. Norvig, *Artificial intelligence : a modern approach*. Pearson, 2016, ISBN: 978-1-292-15396-4.

[5] W. L. Goh and K. T. Lau, "ESMP—expert system for medicine prescription," *Expert Systems with Applications*, vol. 3, no. 4, pp. 457–462, Jan. 1, 1991, ISSN: 0957-4174. DOI: 10.1016/0957-4174(91)90171-A.

[6] W. van Melle, "MYCIN: A knowledge-based consultation program for infectious disease diagnosis," *International Journal of Man-Machine Studies*, vol. 10, no. 3, pp. 313–322, May 1, 1978, ISSN: 0020-7373. DOI: 10.1016/S0020-7373(78)80049-2.

[7] N. McCauley and M. Ala, "The use of expert systems in the healthcare industry," *Information & Management*, vol. 22, no. 4, pp. 227–235, Apr. 1, 1992, ISSN: 0378-7206. DOI: 10.1016/0378-7206(92)90025-B.

[8] M. Mitchell, "Why AI is harder than we think," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '21, New York, NY, USA: Association for Computing Machinery, Jun. 26, 2021, p. 3, ISBN: 978-1-4503-8350-9. DOI: 10.1145/3449639.3465421.

[9] M. Mitchell, *Why AI is harder than we think*, Apr. 28, 2021. DOI: 10.48550/arXiv.2104.12871. arXiv: 2104.12871.

[10] W. J. Clancey, "The epistemology of a rule-based expert system —a framework for explanation," *Artificial Intelligence*, vol. 20, no. 3, pp. 215–251, May 1, 1983, ISSN: 0004-3702. DOI: 10.1016/0004-3702(83)90008-5.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer, 2006, ISBN: 0387310738.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc., 2012.

[13] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012, Conference Name: IEEE Signal Processing Magazine, ISSN: 1558-0792. DOI: 10.1109/MSP.2012.2205597.

[14] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. DOI: 10.3115/v1/D14-1181.

[15] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Toutanova and H. Wu, Eds., Baltimore, Maryland, USA: Association for Computational Linguistics, Jun. 2014, pp. 655–665. DOI: 10.3115/v1/P14-1062.

[16] R. Co Klaus A. Kuhnllobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. 76, pp. 2493–2537, 2011.

[17] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, J. Tsujii and J. Hajic, Eds., Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 2335–2344.

[18] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: Promise and potential," *Health Information Science and Systems*, vol. 2, no. 1, p. 3, Feb. 7, 2014, ISSN: 2047-2501. DOI: 10.1186/2047-2501-2-3.

[19] S. Dash, S. K. Shakyawar, M. Sharma, and S. Kaushik, "Big data in healthcare: Management, analysis and future prospects," *Journal of Big Data*, vol. 6, no. 1, p. 54, Jun. 19, 2019, ISSN: 2196-1115. DOI: 10.1186/s40537-019-0217-0.

[20] S. M. Meystre, G. K. Savova, K. C. Kipper-Schuler, and J. F. Hurdle, "Extracting information from textual documents in the electronic health record: A review of recent research," *Yearbook of Medical Informatics*, vol. 17, no. 1, pp. 128–144, 2008, Publisher: Georg Thieme Verlag KG, ISSN: 0943-4747, 2364-0502. DOI: 10.1055/s-0038-1638592.

[21] P. Yadav, M. Steinbach, V. Kumar, and G. Simon, "Mining electronic health records (EHRs): A survey," *ACM Comput. Surv.*, vol. 50, no. 6, 85:1–85:40, Jan. 3, 2018, ISSN: 0360-0300. DOI: 10.1145/3127881.

[22] I. Spasic and G. Nenadic, "Clinical text data in machine learning: Systematic review," *JMIR Medical Informatics*, vol. 8, no. 3, e17984, Mar. 31, 2020. DOI: 10.2196/17984.

[23] G. M. Weber, K. D. Mandl, and I. S. Kohane, "Finding the missing link for big biomedical data," *JAMA*, vol. 311, no. 24, pp. 2479–2480, Jun. 25, 2014, ISSN: 0098-7484. DOI: 10.1001/jama.2014.4228.

[24] F. Prasser, O. Kohlbacher, U. Mansmann, B. Bauer, and K. A. Kuhn, "Data integration for future medicine (DIFUTURE)," *Methods of Information in Medicine*, vol. 57, e57–e65, S 01 Jul. 2018, ISSN: 2511-705X. DOI: 10.3414/ME17-02-0022.

[25] H.-U. Prokosch *et al.*, "MIRACUM: Medical informatics in research and care in university medicine," *Methods of Information in Medicine*, vol. 57, e82–e91, S 01 Jul. 2018, ISSN: 2511-705X. DOI: 10.3414/ME17-02-0025.

[26] A. Winter *et al.*, "Smart medical information technology for healthcare (SMITH)," *Methods of Information in Medicine*, vol. 57, e92–e105, S 01 Jul. 2018, ISSN: 2511-705X. DOI: 10.3414/ME18-02-0004.

[27] B. Haarbrandt *et al.*, "HiGHmed - an open platform approach to enhance care and research across institutional boundaries," *Methods of Information in Medicine*, vol. 57, e66–e81, S 01 Jul. 2018, ISSN: 2511-705X. DOI: 10.3414/ME18-02-0002.

[28] H. Dalianis, M. Hassel, and S. Velupillai, "The stockholm EPR corpus-characteristics and some initial findings," *Proceedings of ISHIMR*, pp. 243–249, 2009.

[29] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, no. 1, pp. 44–53, Jan. 1, 2018, ISSN: 2095-5138. DOI: 10.1093/nsr/nwx106.

[30] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 50–70, Jan. 2022, Conference Name: IEEE Transactions on Knowledge and Data Engineering, ISSN: 1558-2191. DOI: 10.1109/TKDE.2020.2981314.

[31] C. Safran *et al.*, "Toward a national framework for the secondary use of health data: An american medical informatics association white paper," *Journal of the American Medical Informatics Association : JAMIA*, vol. 14, no. 1, pp. 1–9, 2007, ISSN: 1067-5027. DOI: 10.1197/jamia.M2273.

[32] F. Borchert *et al.*, "GGPONC 2.0 - the german clinical guideline corpus for oncology: Curation workflow, annotation policy, baseline NER taggers," in *Proceedings of the Language Resources and Evaluation Conference*, Marseille, France: European Language Resources Association, Jun. 2022, pp. 3650–3660.

[33] P. Richter-Pechanski *et al.*, "A distributable german clinical corpus containing cardiovascular clinical routine doctor's letters," *Scientific Data*, vol. 10, no. 1, p. 207, Apr. 14, 2023, ISSN: 2052-4463. DOI: 10.1038/s41597-023-02128-9.

[34] M. Kittner *et al.*, "Annotation and initial evaluation of a large annotated german oncological corpus," *JAMIA Open*, vol. 4, no. 2, ooab025, Apr. 1, 2021, ISSN: 2574-2531. DOI: 10.1093/jamiaopen/ooab025.

[35] J. J. Berman, "Confidentiality issues for medical data miners," *Artificial Intelligence in Medicine*, Medical Data Mining and Knowledge Discovery, vol. 26, no. 1, pp. 25–36, Sep. 1, 2002, ISSN: 0933-3657. DOI: 10.1016/S0933-3657(02)00050-7.

[36] A. E. W. Johnson *et al.*, "MIMIC-III, a freely accessible critical care database," *Scientific Data*, vol. 3, no. 1, p. 160 035, May 24, 2016, ISSN: 2052-4463. DOI: 10.1038/sdata.2016.35.

[37] A. E. W. Johnson *et al.*, "MIMIC-IV, a freely accessible electronic health record dataset," *Scientific Data*, vol. 10, no. 1, p. 1, Jan. 3, 2023, ISSN: 2052-4463. DOI: 10.1038/s41597-022-01899-x.

[38] S. Ishihara, "Training data extraction from pre-trained language models: A survey," in *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, A. Ovalle *et al.*, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 260–275. DOI: 10.18653/v1/2023.trustnlp-1.23.

[39] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, ISSN: 2375-1207, May 2019, pp. 739–753. DOI: 10.1109/SP.2019.00065.

[40] N. Carlini *et al.*, "Extracting training data from large language models," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2633–2650.

[41] W. W. Chapman, P. M. Nadkarni, L. Hirschman, L. W. D'Avolio, G. K. Savova, and O. Uzuner, "Overcoming barriers to NLP for clinical text: The role of shared tasks and the need for additional creative solutions," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 540–543, Sep. 1, 2011, ISSN: 1067-5027. DOI: 10.1136/amiajnl-2011-000465.

[42] T. Zesch and J. Bewersdorff, "German medical natural language processing – a data-centric survey," *Applications in Medicine and Manufacturing*, vol. 137, 2022.

[43] L. Seiffe, O. Marten, M. Mikhailov, S. Schmeier, S. Möller, and R. Roller, "From witch's shot to music making bones - resources for medical laymen to technical language and vice versa," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, N. Calzolari *et al.*, Eds., Marseille, France: European Language Resources Association, May 2020, pp. 6185–6192, ISBN: 979-10-95546-34-4.

[44] C. Lohr, S. Buechel, and U. Hahn, "Sharing copies of synthetic clinical corpora without physical distribution — a case study to get around IPRs and privacy constraints featuring the german JSYNCC corpus," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, N. Calzolari *et al.*, Eds., Miyazaki, Japan: European Language Resources Association (ELRA), May 2018.

[45] T. Seddig, P. Daumke, J. Paetzold, and K. Markó, "Entwicklung einer NLP-pipeline für die deutsche medizinische literatur," in *53. Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie e. V. (GMDS)*, Stuttgart, Germany: German Medical Science GMS Publishing House, Sep. 10, 2008, DocMI21–5.

[46] J. Wermter and U. Hahn, "An annotated german-language medical text corpus as language resource," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, M. T. Lino, M. F. Xavier, F. Ferreira, R. Costa, and R. Silva, Eds., Lisbon, Portugal: European Language Resources Association (ELRA), May 2004.

[47] M. D. Wilkinson *et al.*, "The FAIR guiding principles for scientific data management and stewardship," *Scientific Data*, vol. 3, no. 1, p. 160 018, Mar. 15, 2016, ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18.

[48] A. Névéol, H. Dalianis, S. Velupillai, G. Savova, and P. Zweigenbaum, "Clinical natural language processing in languages other than english: Opportunities and challenges," *Journal of Biomedical Semantics*, vol. 9, no. 1, p. 12, Mar. 30, 2018, ISSN: 2041-1480. DOI: 10.1186/s13326-018-0179-8.

[49] A. Shaitarova, J. Zaghir, A. Lavelli, M. Krauthammer, and F. Rinaldi, "Exploring the latest highlights in medical natural language processing across multiple languages: A survey," *Yearbook of Medical Informatics*, vol. 32, no. 1, pp. 230–243, Aug. 2023, ISSN: 0943-4747, 2364-0502. DOI: 10.1055/s-0043-1768726.

[50] R. Roller *et al.*, "Information extraction models for german clinical text," in *2020 IEEE International Conference on Healthcare Informatics (ICHI)*, ISSN: 2575-2634, Nov. 2020, pp. 1–2. DOI: 10.1109/ICHI48887.2020.9374385.

[51] A. R. Aronson and F.-M. Lang, "An overview of MetaMap: Historical perspective and recent advances," *Journal of the American Medical Informatics Association*, vol. 17, no. 3, pp. 229–236, 2010, Publisher: BMJ Group BMA House, Tavistock Square, London, WC1H 9JR.

[52] G. K. Savova *et al.*, "Mayo clinical text analysis and knowledge extraction system (cTAKES): Architecture, component evaluation and applications," *Journal of the American Medical Informatics Association : JAMIA*, vol. 17, no. 5, pp. 507–513, 2010, ISSN: 1067-5027. DOI: 10.1136/jamia.2009.001560.

[53] C.-H. Wei *et al.*, "PubTator 3.0: An AI-powered literature resource for unlocking biomedical knowledge," *Nucleic Acids Research*, vol. 52, W540–W546, W1 Jul. 5, 2024, ISSN: 0305-1048. DOI: 10.1093/nar/gkae235.

[54] F. Liu, E. Shareghi, Z. Meng, M. Basaldella, and N. Collier, "Self-alignment pretraining for biomedical entity representations," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova *et al.*, Eds., Association for Computational Linguistics, Jun. 2021, pp. 4228–4238. DOI: 10.18653/v1/2021.naacl-main.334.

[55] H. Eyre *et al.*, "Launching into clinical space with medspaCy: A new clinical text processing toolkit in python," *AMIA Annu Symp Proc*, vol. 2021, pp. 438–447, Jun. 14, 2021.

[56] M. Neumann, D. King, I. Beltagy, and W. Ammar, "ScispaCy: Fast and robust models for biomedical natural language processing," in *Proceedings of the 18th BioNLP Workshop and Shared Task*, D. Demner-Fushman, K. B. Cohen, S. Ananiadou, and J. Tsujii, Eds., Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 319–327. DOI: 10.18653/v1/W19-5034.

[57] R. Leaman, R. Islamaj Doğan, and Z. Lu, "DNorm: Disease name normalization with pairwise learning to rank," *Bioinformatics*, vol. 29, no. 22, pp. 2909–2917, Nov. 15, 2013, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btt474.

[58] F. Liu, I. Vulić, A. Korhonen, and N. Collier, "Learning domain-specialised representations for cross-lingual biomedical entity linking," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Association for Computational Linguistics, Aug. 2021, pp. 565–574. DOI: 10.18653/v1/2021.acl-short.72.

[59] Y. Peng, Q. Chen, and Z. Lu, "An empirical study of multi-task learning on BERT for biomedical text mining," in *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, D. Demner-Fushman, K. B. Cohen, S. Ananiadou, and J. Tsujii, Eds., Association for Computational Linguistics, Jul. 2020, pp. 205–214. DOI: 10.18653/v1/2020.bionlp-1.22.

[60] E. Alsentzer *et al.*, "Publicly available clinical BERT embeddings," in *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, A. Rumshisky, K. Roberts, S. Bethard, and T. Naumann, Eds., Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 72–78. DOI: 10.18653/v1/W19-1909.

[61] J. Lee *et al.*, "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, Feb. 15, 2020, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz682.

[62] L. Rasmy, Y. Xiang, Z. Xie, C. Tao, and D. Zhi, "Med-BERT: Pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction," *npj Digital Medicine*, vol. 4, no. 1, pp. 1–13, May 20, 2021, ISSN: 2398-6352. DOI: 10.1038/s41746-021-00455-y.

[63] K. K. Bressem *et al.*, "medBERT.de: A comprehensive german BERT model for the medical domain," *Expert Systems with Applications*, vol. 237, p. 121 598, Mar. 1, 2024, ISSN: 0957-4174. DOI: 10.1016/j.eswa.2023.121598.

[64] F. Borchert *et al.*, "GGPONC: A corpus of german medical text with rich metadata based on clinical practice guidelines," in *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, 2020, pp. 38–48.

[65] J. A. Kors, S. Clematide, S. A. Akhondi, E. M. van Mulligen, and D. Rebholz-Schuhmann, "A multilingual gold-standard corpus for biomedical concept recognition: The mantra GSC," *Journal of the American Medical Informatics Association*, vol. 22, no. 5, pp. 948–956, Sep. 1, 2015, ISSN: 1067-5027. DOI: 10.1093/jamia/ocv037.

[66] L. Campillos, L. Deléger, C. Grouin, T. Hamon, A.-L. Ligozat, and A. Névéol, "A french clinical corpus with comprehensive semantic annotations: Development of the medical entity and relation LIMSI annOtated text corpus (MERLOT)," *Language Resources and Evaluation*, vol. 52, no. 2, pp. 571–601, Jun. 1, 2018, ISSN: 1574-0218. DOI: 10.1007/s10579-017-9382-y.

[67] R. Leaman, R. Khare, and Z. Lu, "Challenges in clinical natural language processing for automated disorder normalization," *Journal of Biomedical Informatics*, vol. 57, pp. 28–37, Oct. 2015, ISSN: 15320464. DOI: 10.1016/j.jbi.2015.07.010.

[68] O. Patterson, S. Igo, and J. F. Hurdle, "Automatic acquisition of sublanguage semantic schema: Towards the word sense disambiguation of clinical narratives," *AMIA Annual Symposium Proceedings*, vol. 2010, pp. 612–616, 2010, ISSN: 1942-597X.

[69] D. Demner-Fushman, W. W. Chapman, and C. J. McDonald, "What can natural language processing do for clinical decision support?" *Journal of biomedical informatics*, vol. 42, no. 5, pp. 760–772, Oct. 2009, ISSN: 1532-0464. DOI: 10.1016/j.jbi.2009.08.007.

[70] M. Kreuzthaler and S. Schulz, "Disambiguation of period characters in clinical narratives," in *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*, S. Velupillai, M. Duneld, M. Kvist, H. Dalianis, M. Skeppstedt, and A. Henriksson, Eds., Gothenburg, Sweden: Association for Computational Linguistics, Apr. 2014, pp. 96–100. DOI: 10.3115/v1/W14-1115.

[71] M. Kreuzthaler, M. Oleynik, A. Avian, and S. Schulz, "Unsupervised abbreviation detection in clinical narratives," in *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*, A. Rumshisky, K. Roberts, S. Bethard, and T. Naumann, Eds., Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 91–98.

[72] S. Sohn *et al.*, "Clinical documentation variations and NLP system portability: A case study in asthma birth cohorts across institutions," *Journal of the American Medical Informatics Association*, vol. 25, no. 3, pp. 353–359, Mar. 1, 2018, ISSN: 1527-974X. DOI: 10.1093/jamia/ocx138.

[73] T. Brown *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901.

[74] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Comput. Surv.*, vol. 55, no. 9, 195:1–195:35, Jan. 16, 2023, ISSN: 0360-0300. DOI: 10.1145/3560815.

[75] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35, Curran Associates, Inc., 2022, pp. 27 730–27 744.

[76] S. Hua, S. Jin, and S. Jiang, "The limitations and ethical considerations of ChatGPT," *Data Intelligence*, vol. 6, no. 1, pp. 201–239, Feb. 1, 2024, ISSN: 2641-435X. DOI: 10.1162/dint_a_00243.

[77] A. S. Luccioni, S. Viguier, and A.-L. Ligozat, "Estimating the carbon footprint of BLOOM, a 176b parameter language model," *J. Mach. Learn. Res.*, vol. 24, no. 1, 253:11990–253:12004, Mar. 6, 2024, ISSN: 1532-4435.

[78] A. Spirling, "Why open-source generative AI models are an ethical way forward for science," *Nature*, vol. 616, no. 7957, pp. 413–413, Apr. 18, 2023. DOI: 10.1038/d41586-023-01295-4.

[79] J. Liu and H. V. Jagadish, "Institutional efforts to help academic researchers implement generative AI in research," *Harvard Data Science Review*, Special Issue 5 May 31, 2024, Publisher: The MIT Press, ISSN: 2644-2353, 2688-8513. DOI: 10.1162/99608f92.2c8e7e81.

[80] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[81] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "LLM.int8(): 8-bit matrix multiplication for transformers at scale," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22, Red Hook, NY, USA: Curran Associates Inc., Apr. 3, 2024, pp. 30 318–30 332, ISBN: 978-1-7138-7108-8.

[82] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, 140:5485–140:5551, Jan. 1, 2020, ISSN: 1532-4435.

[83] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 2003, pp. 142–147.

[84] F. Stahlberg, "Neural machine translation: A review," *Journal of Artificial Intelligence Research*, vol. 69, pp. 343–418, 2020.

[85] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing," 2017.

[86] H. Touvron *et al.*, *LLaMA: Open and efficient foundation language models*, Feb. 27, 2023. DOI: 10.48550/arXiv.2302.13971. arXiv: 2302.13971[cs].

[87] H. Touvron *et al.*, *Llama 2: Open foundation and fine-tuned chat models*, Jul. 19, 2023. DOI: 10.48550/arXiv.2307.09288. arXiv: 2307.09288[cs].

[88] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Erk and N. A. Smith, Eds., Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162.

[89] Y. Wu *et al.*, *Google's neural machine translation system: Bridging the gap between human and machine translation*, Oct. 8, 2016. DOI: 10.48550/arXiv.1609.08144. arXiv: 1609.08144[cs].

[90] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, E. Blanco and W. Lu, Eds., Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. DOI: 10.18653/v1/D18-2012.

[91] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, vol. 28, Curran Associates, Inc., 2015.

[92] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1, 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

[93] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994, Conference Name: IEEE Transactions on Neural Networks, ISSN: 1941-0093. DOI: 10.1109/72.279181.

[94] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on Machine Learning*, S. Dasgupta and D. McAllester, Eds., ser. Proceedings of Machine Learning Research, vol. 28, Atlanta, Georgia, USA: PMLR, Jun. 17, 2013, pp. 1310–1318.

[95] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[96] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[97] Y. Liu *et al.*, *RoBERTa: A robustly optimized BERT pretraining approach*, Jul. 26, 2019. DOI: 10.48550/arXiv.1907.11692. arXiv: 1907.11692[cs].

[98] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[99] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[100] B. Wang and A. Komatsuzaki, *GPT-j-6b: A 6 billion parameter autoregressive language model*, May 2021. [Online]. Available: https://github.com/kingoflolz/mesh-transformer-jax.

[101] S. Black *et al.*, "GPT-NeoX-20b: An open-source autoregressive language model," in *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, A. Fan, S. Ilic, T. Wolf, and M. Gallé, Eds., Association for Computational Linguistics, May 2022, pp. 95–136. DOI: 10.18653/v1/2022.bigscience-1.9.

[102] L. Ramshaw and M. Marcus, "Text chunking using transformation-based learning," in *Third Workshop on Very Large Corpora*, 1995.

[103] T. Kudo and Y. Matsumoto, "Chunking with support vector machines," in *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001.

[104] K. Uchimoto, Q. Ma, M. Murata, H. Ozaku, and H. Isahara, "Named entity extraction based on a maximum entropy model and transformation rules," in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong: Association for Computational Linguistics, Oct. 2000, pp. 326–335. DOI: 10.3115/1075218.1075260.

[105] E. F. Tjong Kim Sang, "Memory-based named entity recognition," in *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002.

[106] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Knight, A. Nenkova, and O. Rambow, Eds., San Diego, California, USA: Association for Computational Linguistics, Jun. 2016, pp. 260–270. DOI: 10.18653/v1/N16-1030.

[107] S. Henry, K. Buchan, M. Filannino, A. Stubbs, and O. Uzuner, "2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records," *Journal of the American Medical Informatics Association : JAMIA*, vol. 27, no. 1, pp. 3–12, Jan. 2020, ISSN: 1067-5027. DOI: 10.1093/jamia/ocz166.

[108] K. Lybarger, M. Yetisgen, and Ö. Uzuner, "The 2022 n2c2/UW shared task on extracting social determinants of health," *Journal of the American Medical Informatics Association*, vol. 30, no. 8, pp. 1367–1378, Aug. 1, 2023, ISSN: 1527-974X. DOI: 10.1093/jamia/ocad012.

[109] E. F. Tjong Kim Sang and S. Buchholz, "Introduction to the CoNLL-2000 shared task chunking," in *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*, 2000.

[110] H. Nakayama, *seqeval: A python framework for sequence labeling evaluation*, Software available from https://github.com/chakki-works/seqeval, 2018. [Online]. Available: https://github.com/chakki-works/seqeval.

[111] T. Wolf *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Q. Liu and D. Schlangen, Eds., Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6.

[112] U. Hahn, "Clinical document corpora—real ones, translated and synthetic substitutes, and assorted domain proxies: A survey of diversity in corpus design, with focus on german text data," *JAMIA Open*, vol. 8, no. 3, ooaf024, Jun. 1, 2025, ISSN: 2574-2531. DOI: 10.1093/jamiaopen/ooaf024.

[113] WHO Collaborating Centre for Drug Statistics Methodology, *Guidelines for atc classification and ddd assignment*, 2025. [Online]. Available: https://atcddd.fhi.no/atc_ddd_index_and_guidelines/guidelines/.

[114] World Health Organization, *ICD-10: International Statistical Classification of Diseases and Related Health Problems, 10th Revision*. World Health Organization, 1992. [Online]. Available: https://icd.who.int/browse10/2019/en.

[115] BfArM (Federal Institute for Drugs and Medical Devices), *Ops: Operation and procedure classification system*, 2024. [Online]. Available: https://www.bfarm.de/EN/Code-systems/Classifications/OPS-ICHI/OPS/_node.html.

[116] SNOMED International, *Snomed ct: Systematized nomenclature of medicine – clinical terms*, July 2024 Release, 2024. [Online]. Available: https://www.snomed.org/.

[117] O. Bodenreider, "The unified medical language system (UMLS): Integrating biomedical terminology," *Nucleic Acids Research*, vol. 32, pp. D267–D270, suppl_1 Jan. 1, 2004, ISSN: 0305-1048. DOI: 10.1093/nar/gkh061.

[118] S. M. Yimam, I. Gurevych, R. Eckart de Castilho, and C. Biemann, "WebAnno: A flexible, web-based and visually supported system for distributed annotations," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, M. Butt and S. Hussain, Eds., Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 1–6.

[119] J.-C. Klie, M. Bugert, B. Boullosa, R. E. d. Castilho, and I. Gurevych, "The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation," in *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, Place: Santa Fe, USA, Association for Computational Linguistics, Jun. 2018, pp. 5–9.

[120] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Commun. ACM*, vol. 57, no. 10, pp. 78–85, Sep. 23, 2014, ISSN: 0001-0782. DOI: 10.1145/2629489.

[121] A. Delpeuch, "OpenTapioca: Lightweight entity linking for wikidata," in *Proceedings of the 1st Wikidata Workshop (Wikidata 2020)*, L.-A. Kaffee, O. Tifrea-Marciuska, E. Simperl, and D. Vrandečić, Eds., ser. CEUR Workshop Proceedings, ISSN: 1613-0073, vol. 2773, CEUR, Nov. 2, 2020.

[122] World Health Organization, *ICD-11: International Classification of Diseases 11th Revision*. World Health Organization, 2019. [Online]. Available: https://icd.who.int/.

[123] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 15, 1995, ISSN: 0885-6125. DOI: 10.1023/A:1022627411411.

[124] M. Ott *et al.*, "Fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, W. Ammar, A. Louis, and N. Mostafazadeh, Eds., Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 48–53. DOI: 10.18653/v1/N19-4009.

[125] N. Ng, K. Yee, A. Baevski, M. Ott, M. Auli, and S. Edunov, "Facebook FAIR's WMT19 news translation task submission," in *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, O. Bojar *et al.*, Eds., Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 314–319. DOI: 10.18653/v1/W19-5333.

[126] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of IBM model 2," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, L. Vanderwende, H. Daumé III, and K. Kirchhoff, Eds., Atlanta, Georgia, USA: Association for Computational Linguistics, Jun. 2013, pp. 644–648.

[127] M. Jalili Sabet, P. Dufter, F. Yvon, and H. Schütze, "SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, T. Cohn, Y. He, and Y. Liu, Eds., Association for Computational Linguistics, Nov. 2020, pp. 1627–1643. DOI: 10.18653/v1/2020.findings-emnlp.147.

[128] Z.-Y. Dou and G. Neubig, "Word alignment by fine-tuning embeddings on parallel corpora," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, P. Merlo, J. Tiedemann, and R. Tsarfaty, Eds., Association for Computational Linguistics, Apr. 2021, pp. 2112–2128. DOI: 10.18653/v1/2021.eacl-main.181.

[129] A. Conneau *et al.*, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Association for Computational Linguistics, Jul. 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747.

[130] B. Chan, S. Schweter, and T. Möller, "German's next language model," in *Proceedings of the 28th International Conference on Computational Linguistics*, D. Scott, N. Bel, and C. Zong, Eds., Barcelona, Spain: International Committee on Computational Linguistics, Dec. 2020, pp. 6788–6796. DOI: 10.18653/v1/2020.coling-main.598.

[131] R. Scheible *et al.*, "GottBERT: A pure german language model," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 21 237–21 250. DOI: 10.18653/v1/2024.emnlp-main.1183.

[132] L. Gao *et al.*, *The pile: An 800gb dataset of diverse text for language modeling*, Dec. 31, 2020. DOI: 10.48550/arXiv.2101.00027. arXiv: 2101.00027[cs].

[133] L. J. Miranda, Á. Kádár, A. Boyd, S. Van Landeghem, A. Søgaard, and M. Honnibal, *Multi hash embeddings in spaCy*, Dec. 19, 2022. DOI: 10.48550/arXiv.2212.09255. arXiv: 2212.09255[cs].

[134] J. Frei, I. Soto-Rey, and F. Kramer, "DrNote: An open medical annotation service," *PLOS Digital Health*, vol. 1, no. 8, e0000086, Aug. 15, 2022, ISSN: 2767-3170. DOI: 10.1371/journal.pdig.0000086.

[135] J. Frei and F. Kramer, "GERNERMED: An open german medical NER model," *Software Impacts*, vol. 11, Feb. 1, 2022, ISSN: 2665-9638. DOI: 10.1016/j.simpa.2021.100212.

[136] J. Frei and F. Kramer, "German medical named entity recognition model and data set creation using machine translation and word alignment: Algorithm development and validation," *JMIR Formative Research*, vol. 7, no. 1, e39077, Feb. 28, 2023. DOI: 10.2196/39077.

[137] J. Frei, L. Frei-Stuber, and F. Kramer, "GERNERMED++: Semantic annotation in german medical NLP through transfer-learning, translation and word alignment," *Journal of Biomedical Informatics*, vol. 147, p. 104 513, Nov. 1, 2023, ISSN: 1532-0464. DOI: 10.1016/j.jbi.2023.104513.

[138] P. Koen, "Pharaoh: A beam search decoder for phrase-based statistical machine translation models," in *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas: Technical Papers*, R. E. Frederking and K. B. Taylor, Eds., Washington, USA: Springer, Sep. 28, 2004, pp. 115–124.

[139] J. Frei and F. Kramer, "Annotated dataset creation through large language models for non-english medical NLP," *Journal of Biomedical Informatics*, vol. 145, p. 104 478, Sep. 1, 2023, ISSN: 1532-0464. DOI: 10.1016/j.jbi.2023.104478.

[140] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020.

[141] J. Frei and F. Kramer, "Creating ontology-annotated corpora from wikipedia for medical named-entity recognition," in *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, D. Demner-Fushman, S. Ananiadou, M. Miwa, K. Roberts, and J. Tsujii, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 570–579. DOI: 10.18653/v1/2024.bionlp-1.47.

[142] M. Fonseca and S. Cohen, "Can large language models follow concept annotation guidelines? a case study on scientific and financial domains," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 8027–8042. DOI: 10.18653/v1/2024.findings-acl.478.

[143] Y. Ahapova, J. Frei, and F. Kramer, "Transformer-based multilabel NER using wikipedia corpora in multiple languages," in *Intelligent Health Systems – From Technology to Data and Knowledge*, IOS Press, 2025, pp. 878–879. DOI: 10.3233/SHTI250488.

[144] M. Naguib, X. Tannier, and A. Névéol, "Few-shot clinical entity recognition in english, french and spanish: Masked language models outperform generative model prompting," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 6829–6852. DOI: 10.18653/v1/2024.findings-emnlp.400.

[145] H. Schäfer, A. Idrissi-Yaghir, P. Horn, and C. Friedrich, "Cross-language transfer of high-quality annotations: Combining neural machine translation with cross-linguistic span alignment to apply NER to clinical texts in a low-resource language," in *Proceedings of the 4th Clinical Natural Language Processing Workshop*, T. Naumann, S. Bethard, K. Roberts, and A. Rumshisky, Eds., Seattle, WA,

USA: Association for Computational Linguistics, Jul. 2022, pp. 53–62. DOI: 10.18653/v1/2022.clinicalnlp-1.6.