EXPLORATION AND ADAPTATION OF LARGE LANGUAGE MODELS FOR SPECIALIZED DOMAINS

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades

Doktorin rerum naturalium

(abgekürzt: Dr. rer. nat.)

genehmigte Dissertation

von Frau
M. Sc. Betty van Aken

geboren am 26.07.1990
in Bremen, Deutschland

2023

Referent: Prof. Dr. techn. Wolfgang Nejdl
Korreferent: Prof. Dr. Alexander Löser
Korreferent: Prof. Dr. Henning Wachsmuth
Vorsitz: Prof. Dr. Ziawasch Abedjan

Tag der Promotion: 14.12.2023

ABSTRACT

Large language models have transformed the field of natural language processing (NLP). Their improved performance on various NLP benchmarks makes them a promising tool—also for the application in specialized domains. Such domains are characterized by highly trained professionals with particular domain expertise. Since these experts are rare, improving the efficiency of their work with automated systems is especially desirable. However, domain-specific text resources hold various challenges for NLP systems. These challenges include distinct language, noisy and scarce data, and a high level of variation. Further, specialized domains present an increased need for transparent systems since they are often applied in high stakes settings. In this dissertation, we examine whether large language models (LLMs) can overcome some of these challenges and propose methods to effectively adapt them to domain-specific requirements.

We first investigate the inner workings and abilities of LLMs and show how they can fill the gaps that are present in previous NLP algorithms for specialized domains. To this end, we explore the sources of errors produced by earlier systems to identify which of them can be addressed by using LLMs. Following this, we take a closer look at how information is processed within Transformer-based LLMs to better understand their capabilities. We find that their layers encode different dimensions of the input text. Here, the contextual vector representation, and the general language knowledge learned during pre-training are especially beneficial for solving complex and multi-step tasks common in specialized domains.

Following this exploration, we propose solutions for further adapting LLMs to the requirements of domain-specific tasks. We focus on the clinical domain, which incorporates many typical challenges found in specialized domains. We show how to improve generalization by integrating different domain-specific resources into our models. We further analyze the behavior of the produced models and propose a behavioral testing framework that can serve as a tool for communication with domain experts. Finally, we present an approach for incorporating the benefits of LLMs while fulfilling requirements such as interpretability and modularity. The presented solutions show improvements in performance on benchmark datasets and in manually conducted analyses with medical professionals.

Our work provides both new insights into the inner workings of pre-trained language models as well as multiple adaptation methods showing that LLMs can be an effective tool for NLP in specialized domains.

Key words *natural language processing, language models, text classification, domain adaptation, explainability*

ZUSAMMENFASSUNG

Große vortrainierte Sprachmodelle haben die automatisierte Sprachverarbeitung (englisch: natural language processing, kurz: NLP) transformiert. Die verbesserten Leistungen in verschiedenen NLP-Benchmarks machen sie zu einem vielversprechenden Werkzeug – auch für den Einsatz in spezialisierten Domänen. Solche Domänen zeichnen sich durch hochqualifizierte Experten mit besonderem Fachwissen aus. Da diese Fachkräfte rar sind, ist es besonders erstrebenswert, die Effizienz ihrer Arbeit mit automatisierten Systemen zu verbessern. Domänenspezifische Textressourcen stellen NLP-Systeme jedoch vor verschiedene Herausforderungen. Zu diesen gehören der Gebrauch von spezifischer Fachsprache, verrauschte und spärliche Daten sowie ein hohes Maß an Varianz. Darüber hinaus werden in spezialisierten Domänen häufig transparente Systeme benötigt. In dieser Dissertation untersuchen wir, ob große Sprachmodelle (englisch: large language models, kurz: LLMs) für die Bewältigung dieser Herausforderungen geeignet sind und präsentieren Methoden, um sie effektiv an Domänen-Anforderungen anzupassen.

Zunächst untersuchen wir die zugrundeliegende Funktionalität und die Fähigkeiten der Modelle und zeigen, worin deren Vorteile gegenüber bisherigen NLP-Algorithmen für spezialisierte Domänen bestehen. Hierzu analysieren wir typische Fehlerquellen und identifizieren mögliche Verbesserungen durch LLMs. Daraufhin werfen wir einen genaueren Blick auf die Prozesse innerhalb Transformer-basierter Sprachmodelle, um deren Funktionsweise besser zu verstehen. Unsere Analyse zeigt, dass die Schichten der Modelle verschiedene Dimensionen des Inputtextes enkodieren. Die kontextualisierte Vektorrepräsentation und das generelle Sprachwissen, das beim Vortrainieren gelernt wurde, sind dabei besonders vorteilhaft für das Lösen komplexer mehrschrittiger Tasks, die in spezialisierten Domänen üblich sind.

Basierend auf diesen Ergebnissen schlagen wir Lösungen für die weitere Anpassung von LLMs an domänenspezifische Anforderungen vor. Wir konzentrieren uns auf die klinische Domäne, die viele typische Herausforderungen spezialisierter Domänen zeigt. Wir integrieren verschiedene domänenspezifische Ressourcen in unsere Modelle und zeigen, dass sich hierdurch deren Generalisierbarkeit verbessert. Des Weiteren analysieren wir das Verhalten der Modelle bezüglich verschiedener Inputs und führen ein Test-Framework ein, das als Werkzeug für die Kommunikation mit Domänenexperten dienen kann. Abschließend stellen wir einen Ansatz vor, um die Vorteile von LLMs zu nutzen und gleichzeitig Anforderungen wie Interpretierbarkeit und Modularität zu erfüllen. Die vorgestellten Lösungen zeigen Leistungsverbesserungen auf Benchmark-Datensätzen und in manuell durchgeführten Analysen mit medizinischem Fachpersonal.

Unsere Arbeit bietet hiermit sowohl neue Einblicke in die Funktionalität von vortrainierten Sprachmodellen als auch verschiedene Methoden zur Modellanpassung, die zeigen, wie LLMs als effektives Werkzeug in spezialisierten Domänen genutzt werden können.

Schlagwörter: *Automatisierte Sprachverarbeitung, Sprachmodelle, Textklassifikation, Domänenanpassung, Erklärbarkeit*

ACKNOWLEDGMENTS

During my doctoral studies, I have had the opportunity to work with and learn from excellent mentors, colleagues, and friends.

First and foremost, I would like to thank my advisor Alexander Löser. He provided invaluable guidance throughout these years and believed in me and my work when I was skeptical. The effort he puts into supporting his students is remarkable and was of tremendous help during my journey of becoming a researcher. He has been a great mentor, and I hope our discussions about NLP and the world continue even after this chapter is completed.

I also want to thank the members of my thesis committee, Wolfgang Nejdl, for supervising this dissertation and introducing me to the excellent research at L3S, and Henning Wachsmuth, for providing helpful feedback and ideas during the writing of this dissertation.

Furthermore, I would like to thank all members of the Data Science + X research center at the Berliner Hochschule für Technik. I especially thank Felix Gers, for plenty of valuable discussions, always filled with great expertise, sincerity, and a slice of humor; Peter Tröger, for providing great technical infrastructure that enabled and improved our research profoundly; Jörn Kreutel for initially encouraging me to consider research as a career path; and Amy Siu, for her precise feedback and her incomparable kindness.

This dissertation would not have been possible without the help of my co-authors and colleagues at DATEXIS. I particularly want to thank Sebastian Arnold, Rudolf Schneider, Torsten Kilias, Julian Risch, Benjamin Winter, Konstantina Lazaridou, Tom Oberhauser, Paul Grundmann, Alexei Figueroa, and Jens-Michalis Papaioannou (in order of appearance). They supported me in countless ways and also ensured that I had a lot of fun during these years.

I also want to express my personal gratitude to my friends and family, especially to my parents, their spouses, my sister Paula, and to Rita and Dittmar, for their support and encouragement. A very special thanks goes to Niclas, whose endless support and love during these years have been my constant source of joy, and I am incredibly fortunate to have him by my side.

Finally, I want to dedicate this dissertation to Romy, who sweetened the writing of the final lines with her adorable presence.

FOREWORD

The work presented in this dissertation has been published at multiple scientific venues, as follows.

Part I is built on the following publications about exploring deep neural networks and language models in particular:

- Betty van Aken, Julian Risch, Ralf Krestel, Alexander Löser. Challenges for Toxic Comment Classification: An In-Depth Error Analysis. Workshop on Abusive Language Online (**ALW**) co-located with **EMNLP**, 2018. (Full Paper) [van Aken et al., 2018]

- Betty van Aken, Benjamin Winter, Felix A. Gers and Alexander Löser. How Does BERT Answer Questions? A Layer-Wise Analysis of Transformer Representations. ACM International Conference on Information and Knowledge Management (**CIKM**), 2019. (Full Paper) [van Aken et al., 2019]

- Betty van Aken, Benjamin Winter, Felix A. Gers and Alexander Löser. VisBERT: Hidden-State Visualizations for Transformers. The Web Conference (**WWW**), 2020. (Demonstration Paper) [van Aken et al., 2020]

Part II describes research towards the adaptation of large language models to the clinical domain published in the following workshop and conference papers:

- Betty van Aken, Ivana Trajanovska, Amy Siu, Manuel Mayrdorfer, Klemens Budde, Alexander Löser. Assertion Detection in Clinical Notes: Medical Language Models to the Rescue? Second Workshop on Natural Language Processing for Medical Conversations (**NLPMC**) co-located with **NAACL**, 2021. (Short Paper) [van Aken et al., 2021b]

- Betty van Aken, Jens-Michalis Papaioannou, Manuel Mayrdorfer, Klemens Budde, Felix Gers, Alexander Löser. Clinical Outcome Prediction from Admission Notes using Self-Supervised Knowledge Integration. Conference of the European Chapter of the Association for Computational Linguistics (**EACL**), 2021. (Full Paper) [van Aken et al., 2021a]

- Betty van Aken, Sebastian Herrmann, Alexander Löser. What Do You See in this Patient? Behavioral Testing of Clinical NLP Models. Clinical Natural Language Processing Workshop (**ClinicalNLP**) co-located with **NAACL**, 2022. (Full Paper) [van Aken et al., 2022a]

- Betty van Aken, Jens-Michalis Papaioannou, Marcel G. Naik, Georgios Eleftheriadis, Wolfgang Nejdl, Felix A. Gers and Alexander Löser. This Patient Looks Like That Patient: Prototypical Networks for Interpretable Diagnosis Prediction from Clinical Text. Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (**AACL**), 2022. (Full Paper) [van Aken et al., 2022b]

The complete list of publications during my Ph.D. studies follows:

Conference Papers

- Betty van Aken, Jens-Michalis Papaioannou, Marcel G. Naik, Georgios Eleftheriadis, Wolfgang Nejdl, Felix A. Gers and Alexander Löser. This Patient Looks Like That Patient: Prototypical Networks for Interpretable Diagnosis Prediction from Clinical Text. Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (**AACL**), 2022. (Full Paper) [van Aken et al., 2022b]

- Jens-Michalis Papaioannou, Paul Grundmann, Betty van Aken, Athanasios Samaras Ilias Kyparissidis, George Giannakoulas, Felix A. Gers, Alexander Löser. Cross-Lingual Knowledge Transfer for Clinical Phenotyping. Language Resources and Evaluation Conference (**LREC**), 2022. (Full Paper) [Papaioannou et al., 2022]

- Betty van Aken, Jens-Michalis Papaioannou, Manuel Mayrdorfer, Klemens Budde, Felix Gers, Alexander Löser. Clinical Outcome Prediction from Admission Notes using Self-Supervised Knowledge Integration. Conference of the European Chapter of the Association for Computational Linguistics (**EACL**), 2021. (Full Paper) [van Aken et al., 2021a]

- Sebastian Arnold, Betty van Aken, Paul Grundmann, Felix A. Gers and Alexander Löser. Learning Contextualized Document Representations for Healthcare Answer Retrieval. The Web Conference (**WWW**), 2020. (Full Paper) [Arnold et al., 2020]

- Betty van Aken, Benjamin Winter, Felix A. Gers and Alexander Löser. How Does BERT Answer Questions? A Layer-Wise Analysis of Transformer Representations. ACM International Conference on Information and Knowledge Management (**CIKM**), 2019. (Full Paper) [van Aken et al., 2019]

Journal Articles

- Marcos Treviso, Tianchu Ji, Ji-Ung Lee, Betty van Aken, Qingqing Cao, Manuel R. Ciosici, Michael Hassid, Kenneth Heafield, Sara Hooker, Pedro H. Martins, André F. T. Martins, Peter Milder, Colin Raffel, Edwin Simpson, Noam Slonim, Niranjan Balasubramanian, Leon Derczynski, Roy Schwartz. Efficient Methods for Natural Language Processing: A Survey. Transactions of the Association for Computational Linguistics (**TACL**), 2023. (Full Paper) [Treviso et al., 2022b]

Workshop Papers

- Betty van Aken, Sebastian Herrmann, Alexander Löser. What Do You See in this Patient? Behavioral Testing of Clinical NLP Models. Clinical Natural Language Processing Workshop (**ClinicalNLP**) co-located with **NAACL**, 2022. (Full Paper) [van Aken et al., 2022a]

- Betty van Aken, Ivana Trajanovska, Amy Siu, Manuel Mayrdorfer, Klemens Budde, Alexander Löser. Assertion Detection in Clinical Notes: Medical Language Models to the Rescue?. Second Workshop on Natural Language Processing for Medical Conversations (**NLPMC**) co-located with **NAACL**, 2021. (Short Paper) [van Aken et al., 2021b]

- Betty van Aken, Julian Risch, Ralf Krestel, Alexander Löser. Challenges for Toxic Comment Classification: An In-Depth Error Analysis. Workshop on Abusive Language Online (**ALW**) co-located with **EMNLP**, 2018. (Full Paper) [van Aken et al., 2018]

Posters, Reports and Demonstration Papers

- Jesse Dodge, Iryna Gurevych, Roy Schwartz, Emma Strubell, Betty van Aken. Efficient and Equitable Natural Language Processing in the Age of Deep Learning (Dagstuhl Seminar 22232). **Dagstuhl Reports**, 2023. (Report) [Dodge et al., 2022]

- Betty van Aken, Benjamin Winter, Felix A. Gers and Alexander Löser. VisBERT: Hidden-State Visualizations for Transformers. The Web Conference (**WWW**), 2020. (Demonstration Paper) [van Aken et al., 2020]

- Sünje Paasch-Colberg, Betty van Aken, Christian Strippel, Laura Laugwitz, Alexander Löser, Joachim Trebbe, Martin Emmer. Digging deeper: Extending the Error Analysis of a Hate Speech Algorithm With Information Rich Data. International Conference on Computational Social Science (**IC2S2**), 2020. (Poster Presentation) [Paasch-Colberg et al., 2020]

In Submission

- Ji-Ung Lee, Haritz Puerto, Betty van Aken, Yuki Arase, Jessica Zosa Forde, Leon Derczynski, Andreas Rücklé, Iryna Gurevych, Roy Schwartz, Emma Strubell, Jesse Dodge. Surveying (Dis) Parities and Concerns of Compute Hungry NLP Research. 2023. (Full Paper) [Lee et al., 2023]

- Jens-Michalis Papaioannou, Sebastian Jäger, Betty van Aken, Keno K. Bressem, Felix Gers, Felix Biessmann, Alexander Löser. Conformal ProtoPatient: Towards Trustworthy Clinical Outcome Predictions. 2023. (Full Paper)

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Natural language processing (NLP) has become a part of our daily lives: NLP systems translate texts, search the web based on our queries, and grammar-check emails. In recent years, particular progress has been made with systems based on Deep Learning. Such systems learn patterns typically from a large number of data points. Since such large datasets do not exist for many tasks and domains, recent NLP research and application have strongly benefitted from the paradigm of Transfer Learning [Ruder et al., 2019]. Here, the idea is to (pre-)train models on large text corpora and transfer the learned parameters to domains or tasks with fewer available data. Since most textual data sources are unlabeled, pre-training these models is usually done in a self-supervised fashion by introducing auxiliary tasks. Language modeling, which is one of these tasks, has been shown to be especially effective for Transfer Learning [Devlin et al., 2019]. In this dissertation, we examine the abilities of the resulting pre-trained language models and present different approaches to adapting them to specialized domains.

The last years have shown progress in NLP in both general and domain-specific languages. However, the application in highly specialized domains, such as the clinical domain, still involves many challenges. Specialized domains are characterized by highly trained professionals with domain expertise that far exceeds common knowledge. Because these trained professionals are usually rare and their time limited, supporting their work with automated systems that improve efficiency is highly desirable. In this dissertation, we show directions on how to utilize large language models for this purpose.

1.1.1 NLP for Specialized Domains

One attribute of specialized domains is their highly educated and specialized workforce. The last two decades have shown many efforts for digitalization in domains such as the clinical or legal domain. A large amount of this data consists of unstructured textual documents, such as patient progress notes, as part of Electronic Health Records (EHR). The automatic processing of these documents holds great potential: Work can be made more efficient by sharing experiences faster and more precisely and by therefore enhancing decision-making processes.

However, domain-specific text resources are often more difficult to process due to a number of challenges they pose to NLP systems:

1. **Distinct language and vocabulary** Language in specialized domains usually follows distributions that differ strongly from those of general language. This includes syntactical patterns but also the use of specialized vocabulary. Beyond that, we see a difference in the semantics of used words and phrases. For example, the phrase "information" is frequently used in general language but has a different and specific meaning in the legal domain, where it stands for "a formal criminal charge made by a prosecutor" (see Black's Law Dictionary [Garner, 2014]). These differences in syntactical and semantic distributions make the transfer to specialized domains challenging.

2. **Domain knowledge not fully represented in data** One of the characteristics of specialized domains is the requirement for extensive training of domain professionals. Their knowledge and experience in the field are a substantial part of their everyday work. Some of these are transferred into the documents produced during their work and, therefore, learnable by automatic systems. However, a large amount of the acquired domain knowledge is not accessible in this way. Thus, when supporting the work of domain professionals with NLP systems, we often need to additionally incorporate knowledge from multiple sources beyond labeled task data. Additional data sources can also appear in modalities different from text, such as images, tables, or audio. For example, in the clinical domain, such data might include medical scans, audio recordings from doctors or patients, and tabular lab results.

3. **Lack of shareable data / Data silos** In contrast to data in general domains, we often see a lack of shareable data in specialized domains. This can be caused by privacy regulations but also by the business value that companies attribute to their collected data. However, such data silos within institutions hinder the transferability of models and collective knowledge in many fields and pose a challenge to NLP systems dependent on extensive and variant data sources.

4. **Need for explainable solutions** Models trained to solve tasks in specialized domains are often required to learn complex patterns. For state-of-the-art models based on Deep Learning, these learned patterns are usually opaque both to

the end user and the model creator. This poses problems to more sensitive use cases in which decisions have a large impact. Here, higher levels of trust are needed to ensure the models follow the correct patterns and to prevent the so-called "Clever Hans effect" [Heinzerling, 2020] in which performance is wrongly attributed to assumed abilities of the model. Therefore, explaining model predictions and understanding model behavior becomes very important in these high stakes settings [Rudin, 2019].

1.1.2 Large Language Models for Adaptive Text Representations

Today's NLP systems are mostly based on distributed vector representations of text. Finding text representations that incorporate syntactical and semantic meaning is at the core of building functioning NLP systems. Following the paradigm of Transfer Learning, recent research has shown that well-formed text representations can be shared to solve a variety of downstream tasks [Howard and Ruder, 2018]. Most recent and popular approaches to so-called universal text representations are based on pre-trained large language models.

The objective of neural language modeling was first introduced by Bengio et al. [2003]. It incorporates the idea that distributions of words, or tokens, describe essential parts of a language. By learning these distributions, language models are able to represent both paradigmatic and syntagmatic relations [Sahlgren, 2008].

Our vision is to utilize such large language models as adaptive text representations. These representations contain information about general language and should be shapeable to domain-specific language and different task definitions. Many of such tasks require the incorporation of domain knowledge from multiple data sources (e.g. ontologies). We are searching for ways to include such domain data—coming in various shapes—into pre-trained language models without losing previously acquired knowledge, described as *catastrophic forgetting* [McCloskey and Cohen, 1989].

We further need to be aware that LLMs are not always able to fully solve domain- and task-specific requirements. In many cases, we have to understand them as building blocks for more complex systems that we can shape depending on the domain and task at hand. Understanding how LLMs can be adapted to function as building blocks within such systems is one objective of this dissertation.

We divide our research addressing the vision of adaptable text representations from LLMs into two parts. First, we **explore** how large language models can be beneficial in the scenario of highly domain-specific text. To achieve this, we first evaluate common error classes of prior approaches and then analyze how LLMs produce text representations that can circumvent such errors. The second part considers the **adaptation** of LLMs to specialized domains. In particular, we look at the clinical domain, which incorporates many typical characteristics of specialized domains. To

this regard, we examine how NLP systems based on LLMs can be adapted to the clinical language and domain-specific requirements.

1.2 Research Questions

Our goal is to understand how large language models work (*explore*) and what is needed to apply them to text resources within specialized domains (*adapt*). With these objectives, we pose the following research questions (RQ):

RQ1: What are common errors of machine learning models in specialized domains? How can large language models help to address them?

To understand the usefulness of LLMs in specialized domains, we first have to analyze where previous models are failing. There is a variety of deep and shallow machine learning approaches that can be used for text classification or further NLP tasks in specialized domains. Since they function differently, we assume that they have different strengths and weaknesses. Understanding the error classes that all of these approaches share is a requirement to see the gaps that LLMs can potentially fill to move the field forward.

RQ2: How do large language models process information throughout their layers?

Related work (e.g. Devlin et al. [2019]) has shown that LLMs outperform previous deep learning approaches by large margins on a variety of tasks. However, before applying them to sensitive use cases, we require a better understanding of the inner workings of these models. Since LLMs transform input text through multiple layers to form a final representation, we expect that those layers are responsible for different aspects of the text representation. Analyzing how information is stored and processed throughout the layers of LLMs helps us to understand which adaptations are required for using these models in domain-specific scenarios.

RQ3: How can we incorporate domain-specific knowledge into LLMs in the clinical domain?

Pre-trained LLMs follow the paradigm of Transfer Learning. We expect the information about general language use that is encoded within these models to be beneficial for application in specialized domains. However, since domain-specific information is missing from most pre-training corpora, we need to find ways to incorporate such knowledge for use in specialized domains. We require strategies that preserve the base knowledge encoded within pre-trained models while adapting to the differences in language and tasks of domain-specific settings. In this dissertation, we especially focus on the clinical domain and ways to incorporate clinical knowledge from different

sources into the models. These sources include medical ontologies and scientific articles that contain verified medical findings to complement information from individual patient cases.

RQ4: How can we make large language models more transparent to serve domain requirements?

As described in 1.1.1, specialized domains hold multiple challenges for NLP that can differ depending on specific use cases. To use the benefits that large language models provide for language processing, we need to integrate them into systems that meet such domain-specific requirements. One requirement which is ubiquitous in specialized domains is the transparency of models and their predictions. Since this is not inherent to LLMs, that usually work as black boxes, we especially aim to find ways to incorporate them into more interpretable systems. Further, we study different ways of communicating the behavior and the abilities of LLM-based systems to domain experts.

1.3 Contributions

In this dissertation, we approach the challenges of NLP for specialized domains by adapting large language models to domain-specific requirements. To achieve this, we first explore their abilities and then show how to adapt LLMs to the clinical domain. The clinical text domain serves as a representative for specialized domains in our work as it comprises all specified challenges of such domains. We present the outcomes of our research through the following contributions divided into an exploration and adaptation part:

1.3.1 Exploration

The first contributions in this dissertation address research questions 1 and 2 and help to gain a deeper understanding of the abilities and inner workings of LLMs.

Challenges for Domain-Specific Text Representations

- We compare a range of shallow and deep learning classifiers to a domain-specific multi-label dataset of more than 200,000 user comments. Each classifier, such as Logistic Regression, bidirectional Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN) is meant to tackle specific challenges for domain-specific text classification. We apply the same classifiers to a second dataset of Tweets to validate our results on a different domain.
- We compare the classifiers' predictions and show that they make different errors measured by Pearson correlation coefficients and F1 scores. With the goal of

creating an optimal combination of all approaches, we propose an ensemble that outperforms all individual classifiers.

- We perform a detailed error analysis on the results of the ensemble. The analysis highlights common errors in all evaluated approaches. It further shows which characteristics of pre-trained large language models are missing from these previous approaches.

A Layer-Wise Analysis of Transformer Representations (VisBERT)

- We analyze the abilities within layers of large language models and show how they are impacted by fine-tuning. To that end, we apply a set of NLP probing tasks to each layer of pre-trained and fine-tuned BERT models.
- We show that the text transformations go through similar phases, even if fine-tuned on different tasks. Information about general language properties is encoded in earlier layers and implicitly used to solve the downstream task at hand in later layers.
- We further present a layer-wise visualization of token representations that reveals information about the internal state of the networks. We release this visualization as an interactive online tool that allows users to identify the different phases of text representations within LLMs. The tool is available at https://visbert.demo.datexis.com.
- The code to all conducted experiments is available under Apache License 2.0 at https://github.com/bvanaken/explain-BERT-QA.

1.3.2 Adaptation

The following contributions concern the adaptation of LLMs to the domain of clinical text. Hereby, we address research questions 3 and 4 and show how to apply and adapt large language models to domain-specific tasks.

Language Models for Clinical Assertion Detection

- We evaluate medical language models on Assertion Detection in clinical notes and show that they clearly outperform previous baselines. We further study the transferability of such models to clinical text from other medical areas.
- We manually annotate 5,000 assertions for the MIMIC-III Clinical Database [Johnson et al., 2016]. We release the annotations to the research community to tackle the problem of label sparsity and the lack of diversity in existing data.

- We conduct an error analysis to understand the capabilities of the best-performing model on the task and to reveal directions for improvement.
- We make our system publicly available as a web application to allow further analyses at https://ehr-assertion-detection.demo.datexis.com.
- All annotations and the experimental code are available under Apache License 2.0 at https://github.com/bvanaken/clinical-assertion-data. The model weights are released at https://huggingface.co/bvanaken/clinical-assertion-negation-bert.

CORe: Adapting LLMs to Clinical Outcome Prediction

- We present a novel task setup for clinical outcome prediction that simulates the patient's admission state and predicts the outcome of the current admission.
- We introduce self-supervised clinical outcome pre-training, which integrates knowledge about patient outcomes into existing language models.
- We further propose a method that injects hierarchical signals from medical coding ontologies into the models.
- We compare our approaches against multiple baselines and show that they improve performance on four relevant outcome prediction tasks with up to 1,266 classes. We show that the models are transferable by applying them to a second public dataset without additional fine-tuning.
- We present a detailed analysis of our model that includes a manual evaluation of samples conducted by medical professionals.
- The strengths and weaknesses of our model are demonstrated in an online application available at https://outcome-prediction.demo.datexis.com.
- The code to all experiments is available under Apache License 2.0 at https://github.com/bvanaken/clinical-outcome-prediction. We further release the model weights produced with the presented outcome pre-training approach at https://huggingface.co/bvanaken/CORe-clinical-outcome-biobert-v1.

Behavioral Testing of Clinical Language Models

- We introduce a behavioral testing framework specifically for clinical NLP models. The framework is intended for the evaluation of LLM behavior regarding certain patient descriptions in clinical notes.
- We present an analysis of the patient characteristics gender, age, and ethnicity to understand the sensitivity of models regarding textual cues identifying these groups and whether their predictions are medically plausible.

- We show the results of three state-of-the-art clinical NLP models and find that model behavior strongly varies depending on the applied pre-training. We further show that highly optimized models tend to overestimate the effect of certain patient characteristics leading to potentially harmful behavior.
- We release the code for applying and extending the framework to enable in-depth evaluations by researchers and practitioners. It is available under Apache License 2.0 at https://github.com/bvanaken/clinical-behavioral-testing.

ProtoPatient: Interpretable Diagnosis Prediction Using Prototypical Networks and LLMs

- We introduce a novel model architecture that enables interpretable diagnosis prediction. The architecture is based on LLMs, prototypical networks, and label-wise attention. The system learns relevant parts in the text and points towards prototypical patients that have led to a certain decision.
- We compare our model against several state-of-the-art baselines and show that it outperforms earlier approaches. Performance gains are especially visible in rare diagnoses.
- We further evaluate the explanations provided by our model. The quantitative results indicate that our model produces explanations that are more faithful to its inner workings than post-hoc explanations. A manual analysis conducted by medical doctors further shows the helpfulness of prototypical patients during clinical decision-making.
- We publish an interactive demo application showcasing the benefits of the explainable ProtoPatient approach at https://protopatient.demo.datexis.com.
- We release the code for the model and experiments for reproducibility under Apache License 2.0 at https://github.com/bvanaken/ProtoPatient.

1.4 Thesis Outline

The remainder of this thesis is organized as follows:

In Chapter 2, we discuss existing work that lays the foundation for the concepts described in this dissertation. We first give an overview of the development of text representations that lead to today's wide application of language models for this purpose. Secondly, we provide an introduction to how LLMs are applied for text classification, which is one of the fundamental tasks of NLP, and the basis for most systems discussed in this dissertation.

The following chapters (Chapter 3 - 8) provide detailed descriptions of our research. We divide our work into two parts: exploring large language models and adaptation to specialized domains. Besides presenting experimental details and results, we also focus on analyzing errors in the presented approaches to highlight both strengths and weaknesses and to give suggestions for future research.

In the first part, we take a closer look at why large language models provide a promising direction for domain-specific NLP. Chapter 3 discusses common error classes of previous methods, while Chapter 4 analyzes the inner functionality of LLMs and how they are able to solve some of the discussed errors.

The second part focuses on the adaptation of LLMs to the clinical domain. In Chapter 5, we describe how domain-specific LLMs can be beneficial to the task of clinical assertion detection. Chapter 6 then presents two novel self-supervised adaptation strategies for improved alignment of language models to the domain-specific clinical outcome prediction task.

In Chapter 7, we present a framework for analyzing domain-specific characteristics of our models. Results from applying this framework to LLMs indicate the need for more transparent systems to fulfill domain requirements. Chapter 8 then proposes a system that provides enhanced transparency by combining prototypical networks with LLMs and label-wise attention. A manual analysis with medical doctors confirms that this interpretable system is beneficial to the domain-specific use case.

Chapter 9.1 gives a summary of this dissertation and relates the results to the research questions posed in Section 1.2.

Finally, in Chapter 9.3 we give an outlook on future work concerning the application and adaptation of large language models. In this regard, we discuss the integration of further modalities and languages into LLMs. We further address the issue of resource consumption, which will become even more urgent in the coming years, and highlight promising research directions for more efficient use of LLMs.

2 Background

In this chapter, we discuss the foundations of neural text representations, and how large language models evolved as the leading paradigm for representing information stored in textual documents. We further describe how large language models are applied as building blocks for text classification, an essential task for many applications in specialized domains. Note that we introduce related background work regarding domain-specific NLP tasks and requirements individually in the respective chapters of this dissertation.

2.1 Evolution of Text Representations

The statistical models we use to process documents in natural language require numerical inputs. Therefore, a fundamental question in NLP is how to represent words (or tokens) as numbers. The chosen representation has a large impact on the capabilities of our models, as it determines the information density of our input. The way documents are represented in numbers has strongly shifted within the last decade. This section will give an overview of the development of text representations leading to the current wide use of language models.

2.1.1 Distributional Semantics

The field of distributional semantics addresses the question of how to incorporate meaning of linguistic elements, such as words and phrases, into generalized numerical representations [Turney and Pantel, 2010]. However, Harris [1954] points out that there is no "single or central meaning" to these linguistic elements. In this regard, distributional semantics is based on the idea of Wittgenstein and Anscombe [1953] that "meaning is use" and that, therefore, the meaning of a word can be derived from how it is used, i.e., its distribution within a language. The core of the *distributional*

hypothesis is then based on the correlation between semantic similarity and distributional similarity described by Harris. This is further specified by Firth [1957], who stated that a word is characterized "by the company it keeps".

Given that we can calculate the distributional similarity of words and other linguistic elements, the distributional hypothesis allows using the results to estimate semantic similarity [Sahlgren, 2008]. In the following, we describe approaches that use this distributional hypothesis for creating numerical representations from text.

2.1.2 Bag of Words and TF-IDF

Following the concept of distributional semantics, the *bag of words* (BOW) vector space model represents text documents by their word occurrences. In the resulting vector space, we can determine the semantic similarity of two texts by measuring the distance between their vectors. The BOW approach uses a term-document matrix to build vector representations from a corpus D of text documents. Each row corresponds to a unique term t, e.g. a word, and each column to a document d in the corpus. The frequency f of terms in a document then determines the values of the matrix elements. The considered terms, i.e., the vocabulary, can be chosen according to the use case, e.g., all terms occurring in the corpus, the most common ones, or specific words of interest. A row in the matrix then corresponds to the vector representation of one text document.

Extensions to the BOW model use different weightings of the terms depending on their importance in the document. The most common approach, called *TF-IDF*, gives higher scores to terms that are more distinct within a corpus and are, therefore, better representatives of a document. Here, in addition to the term frequency (TF), we calculate the inverse term frequency (IDF) [Jones, 2004] considering the full corpus. The document vectors are then constructed with TF-IDF values per term.

$$TF(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2.1}$$

$$IDF(t,D) = log \frac{|D|}{|\{d \in D : t \in d\}| + 1} \tag{2.2}$$

$$TF\text{-}IDF(t,d,D) = TF(t,d) \cdot IDF(t,D) \tag{2.3}$$

One problem with these approaches is the sparseness of the vector representations. Since the word distribution in natural language follows Zipf's law [Zipf, 1949], the resulting count-based vectors from natural language documents commonly have a large number of zero entries, or, alternatively, the used vocabulary needs to be strongly restricted to reduce sparseness. In addition to that, both BOW and TF-IDF do not encode information about word meanings and their relationships.

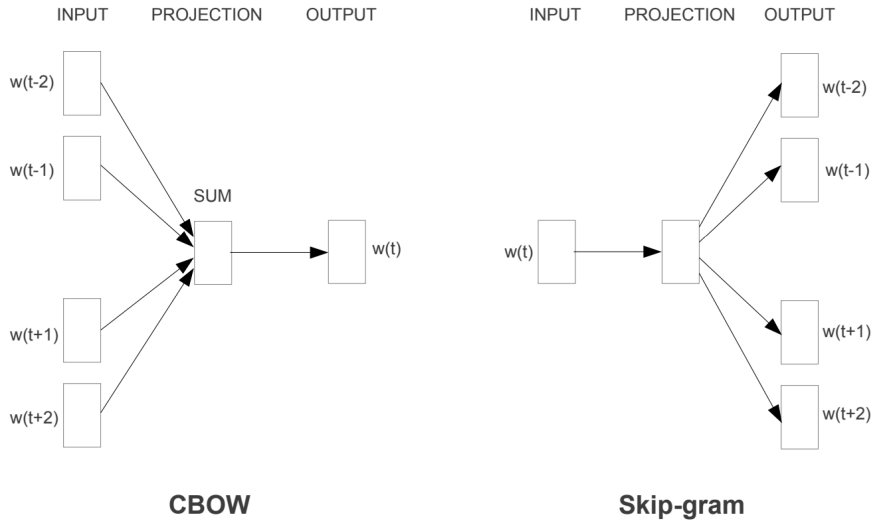


Figure 2.1. Schema of skip-gram and CBOW algorithm. Figure from Mikolov et al. [2013].

2.1.3 Dense Word Embeddings

The one- or multi-hot encodings of the BOW-based vector space model produce sparse vectors. This leads to computation overhead and less efficient parameter usage in the subsequent models. Further, the count-based representations do not encode word meaning and therefore lack important signals about similarities and relationships. These shortcomings lead to research towards dense word embeddings that are computationally efficient and incorporate the semantics of words.

An approach that fulfills both these needs is the use of predictive embedding models. The idea is to use a neural network to determine the embeddings of each word based on an auxiliary prediction task. Mikolov et al. [2013] introduced *Word2Vec* for calculating dense word embeddings, which has become the prevalent technique due to its simplicity and efficiency. Word2Vec incorporates two architectures, the *skip-gram* model and the *continuous bag of words* (CBOW) model. However, the underlying principle is similar in that a model is trained to predict which words occur in a certain context. The Word2Vec approach thus follows the idea of the distributional hypothesis and translates it into training objectives for neural networks.

Given a word w_t, the objective of skip-gram is to maximize the log-likelihood ($\mathcal{L}$) of all surrounding context words C_t. The context window of considered surrounding words is variable.

$$\mathcal{L}_{skip\text{-}gram} = \sum_{t=1}^{T} \sum_{c \in C_t} \log p(w_c | w_t) \tag{2.4}$$

The CBOW architecture follows a similar objective. However, here the task is to predict w_t, given the sum (*bag*) of surrounding context words C_t. Figure 2.1 illustrates both algorithms.

Mikolov et al. [2013] train a neural network with a single hidden layer on this task. The dense word embeddings can then be retrieved from the hidden states of the trained network. In accordance with the objective, words that occur in similar contexts will have similar dense word embeddings, which thus encode paradigmatic relationships.

An extension to the Word2Vec approaches was introduced by Bojanowski et al. [2017]. Their algorithm called *FastText* introduced sub-words as an addition to the word representations. Here, the full representation of a word is calculated as the average of all of its character n-grams. This allows the incorporation of morphology and reduces the problem of out-of-vocabulary words.

2.1.4 Contextualized Word Embeddings

Word embeddings based on pre-trained models, such as Word2Vec, presented large improvements in representing semantic information about words in an efficient way. However, static word embeddings that work in the manner of a lookup table also show weaknesses:

- **Ambiguous words** Static word embeddings cannot handle the ambiguousness of words because they only allow one vector for each word in a vocabulary. Especially if one word meaning is predominant, the resulting word embedding will mainly reflect the most frequent meaning and fail to capture others.
- **Negation** It is difficult to model negation in this setup. Since the representation of the word "great" is unchanged even if there is a preceding negation, the phrase "this is not great" will be strongly influenced by the positive representation of the word "great".
- **Out-of-vocabulary words** Words that are not part of the selected training vocabulary are usually assigned with a generic representation. However, humans are capable of inferring some information about words from context, even though a word is unknown. For example, the sentence "she took the [unkown] out of the shelf and cut it into two halves" already gives clues about the semantics of the unknown word.

Tackling these deficiencies, the idea of contextualized word embeddings is to incorporate the local and global context of words when defining their vector representation. To this end, Peters et al. [2018] propose *Embeddings from Language Models* (ELMo) to enhance word representations with document-level context information.

Their work is built upon the concept of language modeling, first proposed by Shannon [1948]. Language models are trained to predict the next element of a language sequence, which can be a character, (sub-word) token, or word. Bengio et al. [2003] introduced neural probabilistic language modeling using neural networks. Here, the goal was to learn distributed word representations jointly with a probability function for sequences of words.

Peters et al. [2018] use a similar language modeling objective but extend it with a recurrent network architecture that uses character-level input, bidirectional document context, and multiple hidden layers. Further, they focussed their setup on the creation of contextualized word embeddings for improving the performance of downstream tasks. The training objective for ELMo is to maximize the log-likelihood of a token t_k within a sequence $s = (t_1, \dots t_N)$ considering its left and right context.

$$\mathcal{L}_{ELMo} = \sum_{k=1}^{N} \Big(\log p(t_k | t_1 \dots t_{k-1}; \Theta_x, \Theta_{LSTM\rightarrow}, \Theta_s) + \log p(t_k | t_{k+1} \dots t_N; \Theta_x, \Theta_{LSTM\leftarrow}, \Theta_s) \Big)$$

where param x is the jointly learned character-based token representation, Θ_s is the final softmax layer, and $\Theta_{LSTM\rightarrow/\leftarrow}$ describe the parameters of a Recurrent Neural Network (RNN) with *Long Short-Term Memory* (LSTM) [Hochreiter and Schmidhuber, 1997] layers going in both directions. See 3.4.2 for an introduction to LSTM models.

The resulting contextualized word embeddings are then calculated using the hidden states h of the RNN with task-specific weights γ and s.

$$\text{ELMo}_k^{task} = \gamma^{task} \sum_{j=1} s_j^{task} h_{k,j} \tag{2.5}$$

where k is the token index, and j is the hidden layer index. After training, the parameters of the model are frozen and can be used to produce contextualized embeddings. This is done by placing ELMo layers underneath common neural network architectures, replacing static word embeddings such as those produced by Word2Vec. In contrast to static embeddings, ELMo vectors cannot be pre-computed due to their context dependency and are calculated individually for each input. This computational overhead is, however, compensated by significant performance improvements on various downstream tasks such as Question Answering, Semantic Role Labeling, and Named Entity Extraction.

The RNN-based architecture of ELMo has the disadvantage of sequential computation, which precludes parallelization. Therefore, the *Transformer* architecture, mostly composed of attention modules and introduced by Vaswani et al. [2017] has replaced RNN-based language models in recent years. We give an overview of the Transformer architecture in Section 2.2.1.

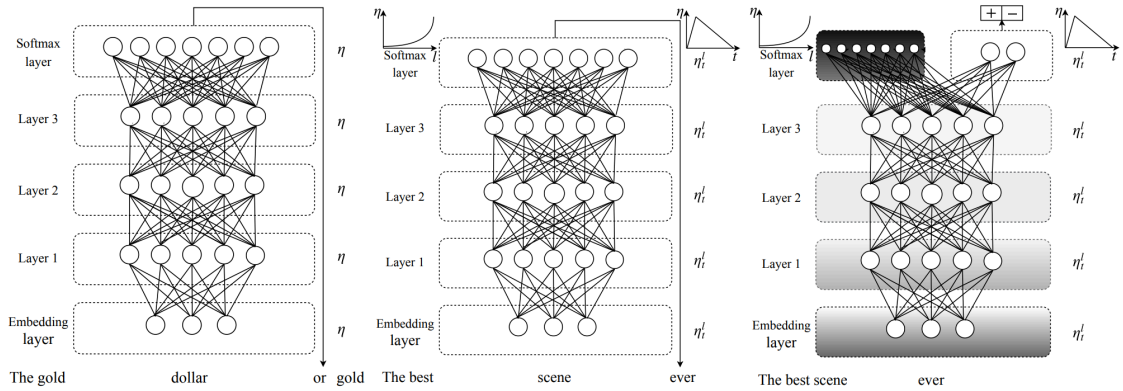


Figure 2.2. Training steps in ULMFit. Figure from Howard and Ruder [2018].

2.1.5 Pre-trained Language Models

The idea to use neural networks for transferring knowledge from a source (usually a large unlabeled corpus) to a target task is known as Neural Transfer Learning [Ruder et al., 2019]. While Word2Vec and ELMo are also based on this paradigm, their focus was mainly on the creation of word embeddings. Howard and Ruder [2018] introduced a more extensive setting in which all network parameters are used to transfer knowledge. They proposed a three-staged setup (see Figure 2.2):

1. **LM pre-training**: A model is pre-trained on general-domain text with a language modeling objective.
2. **LM fine-tuning**: The model is further trained on language modeling on domain- or task-specific data.
3. **Classifier fine-tuning**: A classification head is added to the pre-trained model. The parameters of the classifier are fine-tuned jointly with gradually unfrozen parameters of the language model layers. This final fine-tuning step is done on labeled data from the downstream task.

They showed that this setup outperforms state-of-the-art approaches on multiple text classification tasks and especially reduces the number of required labeled training samples.

The concept of sequential transfer learning was picked up by Devlin et al. [2019], who introduced *Bidirectional Encoder Representations from Transformers* (BERT). Instead of the left-to-right approach of language modeling used in previous work [Peters et al., 2018, Brown et al., 2020], they introduced masked language modeling as pre-training objective, which allows bidirectional language modeling. They additionally use a next sentence prediction task for pre-training.

With BERT, the authors presented a pre-trained language model producing contextualized document encodings that showed to be beneficial to a large number of

Model	Parameters	Dataset Size
BERT [Devlin et al., 2019]	3.4E+0.8	16GB
DistilBERT [Sanh et al., 2019]	6.60E+07	16GB
ALBERT [Lan et al., 2020]	2.23E+08	16GB
XLNet (Large) [Yang et al., 2019]	3.40E+08	126GB
RoBERTa (Large) [Liu et al., 2019c]	3.55E+08	161GB
MegatronLM [Shoeybi et al., 2019]	8.30E+09	174GB
T5-11B [Raffel et al., 2020]	1.10E+10	745GB
Turing-NLG [Rosset, 2020]	1.70E+10	174GB
GPT-3 [Brown et al., 2020]	1.75E+11	570GB
GShard [Lepikhin et al., 2021]	6.00E+11	-
Switch-C [Fedus et al., 2022]	1.57E+12	745GB
OPT-175B [Zhang et al., 2022]	1.75E+11	800 GB
BLOOM [Scao et al., 2022]	1.76E+11	1.6TB
PaLM [Chowdhery et al., 2022]	5.40E+11	3.5TB

Table 2.1. Recent large language models with increasing scale regarding parameters and dataset size. Table adapted from [Bender et al., 2021].

NLP tasks. Due to these performance improvements and its accessibility (the model is publicly available, and application to new tasks is straightforward), BERT quickly became state-of-the-art in many NLP domains and tasks.

The past years have also shown a multitude of BERT variants pre-trained on different domains such as the biomedical [Lee et al., 2020], the legal [Chalkidis et al., 2020], and financial [Araci, 2019]. These domain-specific models usually achieve better results in related tasks than the base model, this appears to be amplified when the models use domain-specific tokenization [Gu et al., 2022]. The cross-institutional sharing of pre-trained BERT-based models through projects such as the Hugging Face model hub[1] was especially important for specialized domains in which the number of labeled data points is often restricted, making pre-trained language models especially beneficial.

2.1.6 Scaling Language Models

Devlin et al. [2019] showed that scaling up the BERT model leads to significant performance improvements also for small-scale tasks. While increasing the number of model parameters of LSTM-based models did not lead to steady performance increases [Melamud et al., 2016], language models based on the Transformer architecture were found to widely share this characteristic (e.g. [Devlin et al., 2019, Brown et al.,

[1]URL to Hugging Face model hub: https://huggingface.co/models

2020, Shoeybi et al., 2019]). Following these findings, the last years have produced a number of Transformer-based BERT successors with ever-growing scale.

Scaling in these scenarios can refer to different variables but often include both the size of the models, usually measured in the number of learnable parameters, and the amount of training data used. Since more parameters require more data for successful training [Zhang et al., 2021a], scaling often considers both the increase of parameters and training data. This is shown in Table 2.1, which lists a number of recent language model releases with a growing number of parameters and dataset sizes.

While scaling language models leads to performance increases on many NLP benchmarks (e.g. the GLUE benchmark [Wang et al., 2019]), there are also downsides to the constant increase in scale. Those include increasing environmental costs [Strubell et al., 2019, Schwartz et al., 2020] and the exclusion of research groups that are equipped with fewer resources. Datasets can also become too large for proper curation leading to the reproduction of harmful content and biases [Bender et al., 2021]. In Chapter 9.3.2 we discuss directions for the efficient use of large language models, which might counteract some of these problems.

Recent developments of generative models that produce coherent text, such as GPT-3 [Brown et al., 2020] and ChatGPT [OpenAI, 2022], bring a new spotlight to extremely large language models. In this dissertation, we focus on language models applied to text classification setups since they currently play a more important role for the use in specialized domains. However, in the coming years, generative models might become more prevalent in typical classification settings (see e.g. [Puri and Catanzaro, 2019]).

2.2 Text Classification Using Large Language Models (LLMs)

In this section, we give an introduction to text classification with large language models, in particular with Transformer-based LLMs, which have shown to be successful for various downstream tasks. We first describe the architecture of Transformer encoders and how they incorporate document-wise context into their encodings. Secondly, we discuss task-specific fine-tuning, which is an essential part of the language model adaptations we later examine in this dissertation.

2.2.1 Transformer Encoders

The original Transformer architecture, as proposed by Vaswani et al. [2017], consists of an encoder and a decoder component. This allows for modeling sequence-to-sequence tasks such as translation. For text classification, as Devlin et al. [2019] show, only the encoder part of the Transformer is required. We follow this setup in the majority

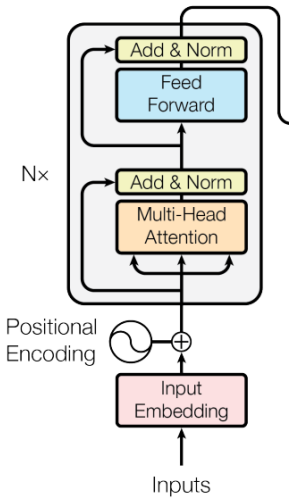


Figure 2.3. Schematic view of the Transformer encoder. Figure from Vaswani et al. [2017].

of work presented in this dissertation and introduce its details in the following.

The Transformer encoder, illustrated in Figure 2.3, is composed of N layers (Devlin et al. [2019] choose $N = 6$ for BERT base and $N = 12$ for BERT large). Each layer contains a multi-head self-attention block and a feed-forward network, with residual connections [He et al., 2016] and layer normalization [Ba et al., 2016] after each part.

Self-attention block. The Transformer does not use recurrent connections as found in LSTMs [Hochreiter and Schmidhuber, 1997] and other recurrent neural networks. Instead, it models relationships between tokens in a pairwise manner via self-attention. The concept of *Attention* was introduced by Bahdanau et al. [2015] for the task of Neural Machine Translation and introduces alignment weights between tokens of a source and a target sequence. These weights determine how much the source tokens contribute to the representation of each target token. Self-attention applies a similar concept to tokens within the same sequence [Parikh et al., 2016, Lin et al., 2017]. Vaswani et al. [2017] formulated their self-attention variation as a mapping of queries and a set of key-value pairs to an output. The output is a weighted sum of the values, where the weights are defined by a compatibility function of the queries and keys. They choose a scaled dot-product as the compatibility function, so that

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.6}$$

where Q, K and V are matrices of queries, keys, and values respectively and $\frac{1}{\sqrt{d_k}}$

denotes the scaling factor. In the Transformer encoder Q, K and V are all constructed from the output of the previous layer.

They further found that performing the attention mechanism multiple times in parallel with differing linear projections allows the model to attend to information from different representation subspaces. Therefore, they introduce multi-head attention, where queries, keys, and values are constructed for each head $i = 1, ..., h$ individually using linear projection matrices W_i^Q, W_i^K, and W_i^V.

$$Q_i = W_i^Q x_{input} \tag{2.7}$$

$$K_i = W_i^K x_{input} \tag{2.8}$$

$$V_i = W_i^V x_{input} \tag{2.9}$$

$$head_i = Attention(Q_i, K_i, V_i) \tag{2.10}$$

The output representations from all heads are then concatenated and multiplied with another parameterized projection matrix W_O.

$$MultiHeadAttention = Concat(head_1, ..., head_h)W^O \tag{2.11}$$

Input representation. The input text for the Transformer encoder is first tokenized and then converted into vectors via word embeddings that are learned end-to-end during training. For tokenization, Devlin et al. [2019] use the *WordPiece* tokenizer [Schuster and Nakajima, 2012, Wu et al., 2016] which is therefore one of the most commonly applied tokenizers for LLMs. However, there are other frequently used tokenization methods such as *Byte Pair Encoding* [Sennrich et al., 2016] which is applied for popular LLMs such as RoBERTa [Liu et al., 2019c] and GPT-2 [Radford et al., 2019].

For applying their BERT model to different text classification tasks, Devlin et al. [2019] propose the use of special tokens. Most importantly, they introduce `[CLS]` as a special token for classifications. It is the first token of each input sequence to BERT. The token's final hidden state is then used as the sequence representation for document-level classification. They further introduce the separator token (`[SEP]`) which indicates a separation between two parts of the text. This is required for tasks that rely on the differentiation between parts of a text, such as Question Answering, in which context and question are denoted using the separator token.

Transformer models do not process input text sequentially but in parallel. Therefore, Vaswani et al. [2017] propose an additional positional encoding to preserve information on token positions within a sequence. From multiple options for positional encodings, they choose a sinusoid function that assigns vectors to tokens depending on their position and is shared across all samples. The positional encoding is then added to the token embeddings as an input to the model layers. Devlin et al. [2019] adopt the positional encoding and enhance it with segment embeddings that identify

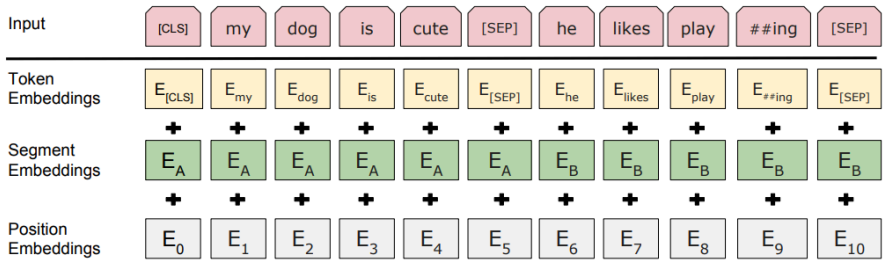


Figure 2.4. Illustration of the input representation for the BERT model. Figure from Devlin et al. [2019].

two parts of an input text (as separated by the `[SEP]` token). The full input to their BERT model is illustrated in Figure 2.4.

2.2.2 Task-specific Fine-tuning

Fine-tuning describes the process of adjusting a pre-trained model to a specific task or domain. As proposed by Howard and Ruder [2018], this is done using a learning rate that is significantly smaller than the one used during the pre-training phase in order to prevent catastrophic forgetting [McCloskey and Cohen, 1989]. When fine-tuning a model on a new task, such as text classification, the common approach is to add a classification head on top of the Transformer encoder layers. The only weights that are added are weights for the classification head. However, instead of adding a classification head, as Devlin et al. [2019] propose, the encoder layers can also be succeeded by more complex architectures. See our proposed ProtoPatient architecture in Chapter 8 for an example of using the Transformer layers as a text encoder building block followed by a prototypical network architecture. In both scenarios, fine-tuning of the Transformer block is done jointly with the learning of the additional weights. Compared to pre-training, fine-tuning is relatively inexpensive since most of the model parameters are already well-adjusted to encode syntactical and semantic information from the input text. This makes fine-tuning of language models on different downstream tasks (e.g. in specialized domains) especially attractive as it allows to efficiently reuse the parameters learned during pre-training.

Prompt engineering. The scaling of recent large language models has led to their development towards few- and zero-shot learners [Brown et al., 2020, Wei et al., 2022]. Kojima et al. [2022], Radford et al. [2019] argue, that many tasks are inherently learned by the models from the large amounts of training data they are exposed to, which makes fine-tuning obsolete. Instead of fine-tuning models to specific tasks, information already stored in the models can, thus, be extracted by using specific prompts, i.e. input sequences causing the model to output results for the requested

task. These findings have stirred a new direction of research concerning the engineering of input prompts to fulfill downstream tasks [Liu et al., 2023]. Since the involved large language models must have a very large number of parameters to perform well without fine-tuning, their use in specialized domains is currently limited by their resource consumption in addition to their black box nature and the corresponding risks in application.

2.3 Chapter Summary

In this chapter, we introduced different approaches for representing natural language text with numerical values. The concept of distributional semantics is leading the efforts of the last decades towards comprehensive representations of textual information. The incorporation of document-wide context into vector embeddings of words and sequences significantly enhanced their expressivity. Language modeling has become the state-of-the-art pre-training paradigm for the creation of such contextualized representations. Especially when pre-trained on a large amount of textual data, the ability of language models to efficiently encode and store information in their parameters makes them useful building blocks for tasks in specialized domains that are often impeded by data sparsity. In the remaining of this dissertation, we explore the abilities of such language models for different domains and tasks. We further show how they can be adapted by task-specific fine-tuning and beyond to address domain-specific challenges and requirements.

Part I

Exploring Large Language Models

3 Challenges for Domain-Specific Text Representations

As discussed in Chapter 2, the wide use of pre-trained large language models has only evolved within the last years. The release of BERT by Devlin et al. [2019] accelerated the wide adoption of this paradigm due to significant performance increases in many downstream NLP tasks. However, since LLMs commonly work as black box models, their underlying mechanisms and capabilities are yet to be fully understood. This understanding is crucial for further improvement and adaptation of these models, particularly in specialized domains and for high stakes use cases. The efforts towards analyzing abilities and inner workings of Transformer-based LLMs has been summarized under the term *BERTology* [Rogers et al., 2020]. In the first part of this dissertation, we present our contributions to this line of research with a focus on specialized domains.

In this chapter, we start by presenting an ensemble-based method for analyzing errors of previous state-of-the-art methods. We use this approach to highlight which challenges are commonly faced when using pre-LLM models in specialized domains and, thus, address research question 1: *What are common errors of machine learning models in specialized domains?* To this end, we take a look at the domain of online conversations and examine automated approaches for moderation support by detecting potentially toxic user comments.

3.1 Introduction

Keeping online conversations constructive and inclusive is a crucial task for platform providers. Automatic classification of toxic comments, such as hate speech, threats, and insults, can help in keeping discussions fruitful. In addition, new regulations in certain European countries have been established enforcing to delete illegal content

within a certain time span.[1]

Research on the topic deals with common NLP challenges, such as long-range dependencies or misspelled and uncommon words. Proposed solutions prior to the use of LLMs include bidirectional Recurrent Neural Networks with attention [Pavlopoulos et al., 2017] and the use of pre-trained static word embeddings [Badjatiya et al., 2017]. However, most pre-existing classifiers tend to fail on the long tail of real world data [Zhang and Luo, 2019]. For understanding the benefits of LLMs for specialized domains, it is essential to know which challenges are already addressed by previous state-of-the-art classifiers and for which they are error-prone.

We take two datasets into account to investigate these errors: comments on Wikipedia Talk Pages presented by Google Jigsaw during Kaggle's Toxic Comment Classification Challenge[2] and a Twitter dataset by Davidson et al. [2017]. These sets include common difficulties in datasets for the task: They are labeled based on different definitions; they include diverse language from user comments and Tweets; and they present a multi-class and a multi-label classification task respectively.

On these datasets, we propose an ensemble of different classifiers. By analyzing false negatives and false positives of the ensemble we collect insights about open challenges that all of the pre-LLM approaches share.

3.2 Related Work

Task definitions. Toxic comment classification is not clearly distinguishable from its related tasks. Besides looking at toxicity of online comments [Wulczyn et al., 2017, Georgakopoulos et al., 2018], related research includes the investigation of hate speech [Badjatiya et al., 2017, Burnap and Williams, 2016, Davidson et al., 2017, Gambäck and Sikdar, 2017, Njagi et al., 2015, Schmidt and Wiegand, 2017, Vigna et al., 2017, Warner and Hirschberg, 2012], online harassment [Yin and Davison, 2009, Golbeck et al., 2017], abusive language [Mehdad and Tetreault, 2016, Park and Fung, 2017], cyberbullying [Dadvar et al., 2013, Dinakar et al., 2012, Hee et al., 2015, Zhong et al., 2016] and offensive language [Chen et al., 2012, Xiang et al., 2012]. Each field uses different definitions for their classification, still similar methods can often be applied to different tasks. In our work, we focus on toxic comment detection and show that the same method can effectively be applied to a hate speech detection task.

Multi-class approaches. Besides traditional binary classification tasks, related work considers different aspects of toxic language, such as racism [Greevy and Smeaton, 2004, Waseem, 2016, Kwok and Wang, 2013] and sexism [Waseem and Hovy, 2016, Jha and Mamidi, 2017], or the severity of toxicity [Davidson et al., 2017, Sharma

[1] https://www.bbc.com/news/technology-42510868
[2] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

et al., 2018]. These tasks are framed as multi-class problems, where each sample is labeled with exactly one class out of a set of multiple classes. The great majority of related research considers only multi-class problems. This is remarkable, considering that in real-world scenarios toxic comment classification can often be seen as a multi-label problem, with user comments fulfilling different predefined criteria at the same time. We, therefore, investigate both a multi-label dataset containing six different forms of toxic language and a multi-class dataset containing three mutually exclusive classes of toxic language.

Shallow classification and neural networks. Toxic comment identification is a supervised classification task and approached by either methods including manual feature engineering [Burnap and Williams, 2015, Mehdad and Tetreault, 2016, Waseem, 2016, Davidson et al., 2017, Nobata et al., 2016, Kennedy et al., 2017, Samghabadi et al., 2017, Robinson et al., 2018] or the use of (deep) neural networks [Ptaszynski et al., 2017, Pavlopoulos et al., 2017, Badjatiya et al., 2017, Vigna et al., 2017, Park and Fung, 2017, Gambäck and Sikdar, 2017]. While in the first case manually selected features are combined into input vectors and directly used for classification, neural network approaches are supposed to automatically learn abstract features above these input features. Neural network approaches appear to be more effective for learning complex patterns [Zhang and Luo, 2019], while feature-based approaches preserve some sort of explainability. In this study, we focus on pre-LLM baselines using both deep neural networks and shallow learners.

Ensemble learning. Burnap and Williams [2015] studied advantages of ensembles of different classifiers. They combined results from three feature-based classifiers. Further, the combination of results from Logistic Regression and a Neural Network has been studied by Gao and Huang [2017], Risch and Krestel [2018]. Zimmerman et al. [2018] investigated ensemble models with different hyper-parameters. In contrast to their work, we combine various model architectures and different word embeddings to understand joint errors of all approaches.

3.3 Datasets and Tasks

The task of toxic comment classification lacks a consistently labeled standard dataset for comparative evaluation [Schmidt and Wiegand, 2017]. While there are a number of annotated public datasets in adjacent fields, such as hate speech [Ross et al., 2017, Gao and Huang, 2017], racism/sexism [Waseem, 2016, Waseem and Hovy, 2016] or harassment [Golbeck et al., 2017] detection, most of them follow different definitions for labeling and therefore often constitute different problems.

Class	# of occurrences
Clean	201,081
Toxic	21,384
Obscene	12,140
Insult	11,304
Identity Hate	2,117
Severe Toxic	1,962
Threat	689

Table 3.1. Class distribution of Wikipedia dataset. The distribution shows a strong class imbalance.

Class	# of occurrences
Offensive	19,190
Clean	4,163
Hate	1,430

Table 3.2. Class distribution of Twitter dataset. The majority class of the dataset consists of offensive Tweets.

3.3.1 Wikipedia Talk Pages Dataset

We analyze a dataset published by Google Jigsaw in December 2017 over the course of the *Toxic Comment Classification Challenge* on Kaggle. It includes 223,549 annotated user comments collected from Wikipedia Talk Pages and is the largest publicly available dataset for the task. The comments were annotated by human raters with the six labels *toxic*, *severe toxic*, *insult*, *threat*, *obscene* and *identity hate*. Comments can be associated with multiple classes at once, which frames the task as a multi-label classification problem. Jigsaw has not published official definitions for the six classes, but they do state that they defined a toxic comment as "a rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion".[3]

The dataset features an unbalanced class distribution, shown in Table 3.1. 201,081 samples fall under the majority class *clear* matching none of the six categories, whereas 22,468 samples belong to at least one of the other classes. While the *toxic* class includes 9.6% of the samples, only 0.3% are labeled as *threat*, marking the smallest class.

Comments were collected from the English Wikipedia and are mostly written in English with some outliers, e.g., in Arabic, Chinese or German language. The domain

[3] https://www.perspectiveapi.com

covered is not strictly locatable, due to various article topics being discussed. Still it is possible to apply a simple categorization of comments, as follows[4]:

1. **Community-related**

   Example:
   "If you continue to vandalize Wikipedia, you will be blocked from editing."

2. **Article-related**

   Example:
   "Dark Jedi Miraluka from the Mid-Rim world of Katarr, Visas Marr is the lone surviving member of her species."

3. **Off-topic**

   Example:
   "== I hate how my life goes today == Just kill me now. . . ."

3.3.2 Twitter Dataset

Additionally, we investigate a dataset introduced by Davidson et al. [2017]. It contains 24,783 Tweets fetched using the Twitter API and annotated by CrowdFlower workers with the labels *hate speech*, *offensive but not hate speech* and *neither offensive nor hate speech*. Table 3.2 shows the class distribution. We observe a strong bias towards the *offensive* class making up 77.4% of the comments caused by sampling Tweets by seed keywords from Hatebase.org. We choose this dataset to show that our method is also applicable to multi-class problems and works with Tweets, which usually have a different structure than other online user comments due to their character limitation.

3.3.3 Common Challenges

We observe three common challenges for natural language processing in both datasets:

- **Long-range dependencies** The toxicity of a comment often depends on expressions made in early parts of the comment. This is especially problematic for longer comments (>50 words) where the influence of earlier parts on the result can vanish.
- **Multi-word phrases** We see many occurrences of multi-word phrases in both datasets. Our algorithms can detect their toxicity only if they can recognize multiple words as a single (typical) hateful phrase.

[4]Disclaimer: This chapter contains examples that may be considered profane, vulgar, or offensive. These contents do not reflect the views of the authors and exclusively serve to explain linguistic research challenges.

- **Out-of-vocabulary words** A common problem for the task—and in specialized domains in general—is the occurrence of words that are not present in the training data. These words include slang or misspellings, but also intentionally obfuscated content.

3.4 Methods and Ensemble

In this section, we introduce pre-LLM baseline methods which we choose with regard to the previously introduced common challenges. We further propose our ensemble learning approach. Its goal is to minimize errors by detecting optimal methods for a given comment.

3.4.1 Logistic Regression

The *Logistic Regression* (LR) [Cox, 1958] algorithm is widely used for binary classification tasks. Unlike deep learning models, it requires manual feature engineering. On the other hand, LR permits obtaining insights about the model, such as observed coefficients. Previous research has investigated different features used with LR and found that n-grams of characters and words are highly indicative features for hate speech detection [Waseem and Hovy, 2016]. Following these findings, we choose word and character n-grams as features for the LR models in our analysis.

3.4.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) interpret a document as a sequence of words or character n-grams. Through recurrent connections the networks can obtain a memory of previous inputs so that the output can be influenced by all prior time steps [Graves, 2012]. Training is done by using the *backpropagation through time* (BPTT) algorithm [Werbos, 1990].

We use four different RNN architectures: A *Long Short-Term Memory* (LSTM) [Hochreiter and Schmidhuber, 1997] model, a bidirectional LSTM, a bidirectional *Gated Recurrent Unit* (GRU) [Cho et al., 2014] architecture and a bidirectional GRU with an additional attention layer. We briefly introduce these architectures in the following.

LSTM. A common problem faced when using RNN models is the *vanishing gradient problem* [Hochreiter et al., 2001], which makes it difficult to incorporate context over long sequences. The LSTM architecture introduced by Hochreiter and Schmidhuber [1997] meets this problem by using multiple gating mechanisms within the units of a network. This way, information from previous time steps can be stored, accessed, and forgotten in a flexible way, which diminishes the problem of vanishing gradients.

In our experiments, we use an LSTM model that takes a sequence of words as input. The words are one-hot encoded and then fed into an embedding layer to get dense vector representations. To increase the robustness of the model, we further add a spatial dropout which randomly masks 10% of the input. The sequence of word embeddings is then processed by an LSTM layer with 128 neurons, followed by another 10% dropout. The output is fed into a dense classification layer. For the multi-label task, this layer uses a sigmoid activation, and for the multi-class task, it uses a softmax activation.

Bidirectional LSTM. While a standard LSTM model processes text from left to right, bidirectional LSTMs consist of two LSTM layers that process the input sequence in both directions—the original sequence of words and the reversed order. The outputs of these two layers are then averaged. This way, bidirectional RNNs can often compensate errors caused by long range dependencies.

Bidirectional GRU. Similar to LSTMs, Gated Recurrent Units [Cho et al., 2014] were introduced to meet the problem of vanishing gradients in RNN models. Since GRUs are made up of only two internal gates controlling the information flow (*reset* and *update*), the authors describe them as a simpler alternative to LSTMs.

In addition to using a bidirectional LSTM network, we include a bidirectional network using GRUs into our study to understand how these two conceptually related approaches differ for the task. As the size for the bidirectional layers we choose 64 neurons and adopt all other parts for the GRU model from our standard LSTM.

Bidirectional GRU with attention layer. The concept of attention in combination with RNNs was introduced by Bahdanau et al. [2015]. Attention weights are meant to learn which parts of a sequence to focus on for building an output representation. Gao and Huang [2017] phrase that "attention mechanisms are suitable for identifying specific small regions indicating hatefulness in long comments". In order to detect these small regions in our comments, we add an attention layer to our bidirectional GRU-based network following the work of Yang et al. [2016].

3.4.3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs), initially used for computer vision [LeCun et al., 1998], are made up of layers of convolution filters sliding over local features such as words or characters. They have shown to be an effective approach for text classification tasks [Collobert et al., 2011]. By intuition, they can detect specific combinations of features, while RNNs can extract orderly information [Zhang and Luo, 2019]. On character level, CNNs can potentially deal with obfuscation of words. For our model, we choose an architecture comparable to the approach of Kim [2014].

3.4.4 (Sub-)Word Embeddings

As described in 2.1.3 and confirmed by Zhang and Luo [2019] for the task of hate speech detection, using word embeddings trained on large corpora can be helpful in order to capture information that is missing from the training data.

Therefore, we apply GloVe embeddings released by [Pennington et al., 2014], that belong to the group of pre-trained dense word embeddings, trained on a corpus of two billion Tweets. In addition, we use sub-word embeddings as introduced by Bojanowski et al. [2017] with the FastText tool. The approach considers substrings of a word to infer its embedding. This is important for learning representations for misspelled, obfuscated or abbreviated words, which are often present in online comments. We train FastText embeddings on 95 million comments from Wikipedia Talk Pages[5]. To that end, we apply the skip-gram method with a context window size of 5 and train for 5 epochs.

3.4.5 Ensemble Learning

We expect that each of the chosen classification approaches has specific strengths and weaknesses. We hypothesize that the networks based on LSTM and GRU layers work well for capturing multi-word phrases and short-range contextual information, but miss dependencies ranging over very long sentences. Bidirectionality and the addition of attention layers can compensate such errors to a certain extent, but could have more problems with very rare patterns. CNNs, on the contrary, might be able to better recognize misspelled words while missing some contextual information.

For these reasons, we propose an ensemble approach, which can learn to choose a combination of classifiers that are most capable for a specific kind of input text. We select a number of features to distinguish different kinds of input, namely length of text, ratio of upper cased characters, non-alphabetical characters, exclamation marks and out-of-vocabulary words (using the GloVe vocabulary as reference). These features are closely related to the characteristics of online user comments.

We then use the set of out-of-fold predictions from the various approaches and train an ensemble classifier with gradient boosting decision trees [Friedman, 2001]. We perform 5-fold cross-validation on the whole setup and use the average of the ensemble predictions on our test set for the final results.

3.5 Experimental Study

We aim to train an ensemble that chooses an optimal combination of classifiers based on a set of comment features. Because the classifiers have different strengths and weaknesses, we expect the ensemble to outperform each individual classifier. Based on

[5] https://figshare.com/articles/Wikipedia_Talk_Corpus/4264973

Model	Wikipedia				Twitter			
	P	R	F1	AUROC	P	R	F1	AUROC
CNN (FastText)	.73	.86	.776	.981	.73	.83	.775	.948
CNN (GloVe)	.70	.85	.748	.979	.72	.82	.769	.945
LSTM (FastText)	.71	.85	.752	.978	.73	.83	.778	.955
LSTM (GloVe)	**.74**	.84	.777	.980	.74	.82	.781	.953
Bidirectional LSTM (FastText)	.71	.86	.755	.979	.72	.84	.775	.954
Bidirectional LSTM (GloVe)	**.74**	.84	.777	.981	.73	**.85**	.783	.953
Bidirectional GRU (FastText)	.72	.86	.765	.981	.72	.83	.773	.955
Bidirectional GRU (GloVe)	.73	.85	.772	.981	.76	.81	.784	.955
Bidirectional GRU Attention (FastText)	**.74**	**.87**	**.783**	**.983**	.74	.83	**.791**	**.958**
Bidirectional GRU Attention (GloVe)	.73	**.87**	.779	**.983**	**.77**	.82	**.790**	.952
Logistic Regression (char-ngrams)	**.74**	.84	.776	.975	.73	.81	.764	.937
Logistic Regression (word-ngrams)	.70	.83	.747	.962	.71	.80	.746	.933
Ensemble	**.74**	**.88**	**.791**	**.983**	.76	.83	**.793**	.953

Table 3.3. Comparison of precision, recall, F1-measure, and AUROC on two datasets. The results show that the ensemble outperforms the individual classifiers in F1-measure. The strongest individual classifier on both datasets is a bidirectional GRU network with attention layer.

results from previous experiments mentioned in Section 3.2 we expect that the state-of-the-art models have a comparable performance and none outperforms the others significantly. This is important because otherwise the ensemble learner constantly prioritizes the outperforming classifier. We test the resulting ensemble on both online comments and Tweets to understand its effectiveness regarding differing language characteristics such as comment length and use of slang words.

3.5.1 Setup

To evaluate the performance of pre-LLM approaches on detecting toxic language, we use the following setup: We compare six methods from Section 3.4. For the neural network approaches we apply two different word embeddings each and for LR we use character and word n-grams as features.

We need binary predictions to calculate precision, recall and the resulting F1-measure. To translate the continuous sigmoid output for the multi-label task (Wikipedia dataset) into binary labels we estimate appropriate threshold values per class. For this purpose we perform a parameter search for the threshold to optimize the F1-measure using the whole training set as validation. In case of the multi-class task (Twitter dataset) the softmax layer makes the parameter search needless, because we can simply take the label with the highest value as the predicted one.

We choose the macro-average F1 measure since it is more indicative than the micro-average F1 for strongly unbalanced datasets [Zhang and Luo, 2019]. For the

multi-label classification we measure macro-precision and -recall for each class separately and average their results to get the F1-measure per classifier. The Area under the Receiver Operating Curve (AUROC) gives us a measurement of classifier performance without the need for a specific threshold. We add it to provide additional comparability of the results.

3.5.2 Correlation Analysis

The ensemble can only be effective when models with comparable performance produce uncorrelated predictions. We, thus, measure the correlation of the predictions of different classifiers. We look at a set of combinations, such as shallow learner combined with a neural net, and inspect their potential for improving the overall prediction. For measuring the disparity of two models we use the Pearson correlation coefficient. The results are shown in Table 3.4.

3.5.3 Experimental Results

As shown in Table 3.3, our ensemble outperforms the strongest individual method on the Wikipedia dataset by approximately one percent F1-measure. We also observe that on both datasets the bidirectional GRU with attention approach outperforms the other individual classifiers. This indicates that the contextualization from the bidirectional layers amplified by the attention layer helps in detecting toxicity in user comments.

We see that the difference in F1 between the best individual classifier and the ensemble is higher on the Wikipedia dataset as on the Twitter dataset. This finding is accompanied by the results in Table 3.4 which show that most classifier combinations present a high correlation on the Twitter dataset and are, therefore, less effective on the ensemble. An explanation for this effect is that the text sequences within the Twitter set show less variance than the ones in the Wikipedia dataset. This might be due to the sampling strategy based on a list of terms, the smaller size of the dataset, and the fewer number of classes. With less variant data one selected classifier for a type of text can be sufficient and even more efficient than an ensemble approach.

As the results in Table 3.4 show, ensembling is especially effective on the sparse classes *threat* (Wikipedia) and *hate* (Twitter). The predictions for these two classes have the weakest correlation. This can be especially useful for strongly imbalanced datasets, which are common in toxic comment classification and related tasks. The results give us further indicators for useful combinations of classifiers. Combining the shallow learner approach with neural networks is highly effective. Contrary to that we see that the different word embeddings used do not lead to strongly differing predictions.

Class	F1		Pearson
	Different word embeddings		
	GRU+G	GRU+FT	
W avg.	.78	.78	.95
W threat	.70	.69	.92
T avg.	.79	.79	.96
T hate	.53	.54	.94
	CNN+G	CNN+FT	
W avg.	.75	.78	.91
W threat	**.67**	**.73**	**.82**
T avg.	.77	.78	.94
T hate	.49	.53	.90
	Different NN architectures		
	CNN	BiGRU Att	
W avg.	**.78**	**.78**	**.85**
W threat	**.73**	**.71**	**.65**
T avg.	.78	.79	.96
T hate	.50	.49	.93
	Shallow learner and NN		
	CNN	LR char	
W avg.	**.78**	**.78**	**.86**
W threat	**.73**	**.74**	**.78**
T avg.	.78	.76	.92
T hate	**.50**	**.51**	**.86**
	BiGRU Att	LR char	
W avg.	**.78**	**.78**	**.84**
W threat	**.71**	**.74**	**.67**
T avg.	.79	.76	.92
T hate	.49	.51	.88
	Character- and word-ngrams		
	LR word	LR char	
W avg.	**.75**	**.78**	**.83**
W threat	**.70**	**.74**	**.69**
T avg.	.75	.77	.94
T hate	.50	.51	.91

Table 3.4. F1-measures and Pearson correlations of different combinations of classifiers. When the pearson score is low and F1 is similar, an ensemble performs best. We see that this appears mostly on the Wikipedia dataset and on the respective minority classes *threat* and *hate*. W: Wikipedia dataset; T: Twitter dataset; G: GloVe embeddings; FT: FastText embeddings; avg.: averaged; NN: Neural Networks.

3.6 Detailed Error Analysis

The ensemble of pre-LLM classifiers still fails to reach F1 scores higher than 0.8. To find out the remaining problems, we perform an extensive error analysis on the result of the ensemble. This way, we can identify potential opportunities for improvements by using LLMs.

We analyse common error classes of our ensemble based on research from Zhang and Luo [2019], Zhang et al. [2018], Qian et al. [2018], Davidson et al. [2017], Schmidt and Wiegand [2017], Nobata et al. [2016]. Moreover, we add additional error classes we encountered during our manual analysis. To address deficits in both precision and recall we inspect false negative and false positive classifications. We focus on error classes with the highest frequency in the observed samples. The occurrence of an error class within a comment is taken to be binary (occurs in comment or not).

We present the results on the *toxic* class of the Wikipedia dataset and the *hate* class of the Twitter dataset. Both classes are of high significance for the task of user comment moderation. Our ensemble results in 1794 false negatives and 1581 false positives for the Wikipedia dataset. We choose 200 random samples out of each set as representatives. For the smaller Twitter dataset we get 55 false negatives and 58 false positives. We perform our analysis on all of these samples.

3.6.1 Error Classes of False Negatives

Doubtful labels. We observe a high number of comments for which we question the original label when taking the respective class definition into account. A common occurrence is actual toxic or hateful content that is cited by the comment's author. Another pattern is the use of potentially toxic words within an explanation or self reproach.

Example:
"No matter how upset you may be there is never a reason to refer to another editor as 'an idiot' "

We find that 23% of sampled comments in the false negatives of the Wikipedia dataset do not fulfill the toxic definition in our view. Taking the hate speech definition of the authors into account, we question 9% of the Twitter dataset samples. For the remaining error classes we only include the comments with undoubtful labels.

Toxicity without swear words. Davidson et al. [2017] phrase the problem that hate speech may not contain hate or swear words at all.

Example:
"she looks like a horse"

50% of Wikipedia dataset samples have no common hate or swear word in them. This makes it the largest error class for the Wikipedia dataset and shows that our classifiers often fail when there are no obvious hateful words present. We observe this in 18% of hate speech comments from the Twitter dataset. It is important to notice that the frequency of swear words is naturally higher within this dataset, because of its sampling method with hateful words as seeds. In many cases the problem is a lack of paradigmatic context. Hence, improved semantic embeddings could mitigate the problem by improving the distinction between different contexts.

Misspelled and rare words. Errors caused by rare or unknown words are reported by Nobata et al. [2016], Zhang and Luo [2019], Qian et al. [2018]. From our observation, they include misspellings, neologisms, obfuscations, abbreviations, and slang words. Even though some of these words appear in the embedding, their frequency may be too low to correctly detect their meaning from the word embeddings.

Example:
"fucc nicca yu pose to be pullin up"

We find rare or unknown words in 30% of examined false negatives from the Wikipedia dataset and in 43% of Twitter dataset samples. This also reflects the common language on Twitter with many slang words, abbreviations and misspellings. One option to circumvent this problem is to train embeddings on larger corpora with even more variant language. However, since online language changes frequently, the incorporation of context could be even more important to identify such cases.

Rhetorical questions. As pointed out by Schmidt and Wiegand [2017], it is common practice to wrap toxic statements online within rhetorical or suggestive questions. We find a number of such comments within the false negatives.

Example:
"have you no brain?!?!"

21% of Wikipedia dataset samples and 10% of Twitter dataset samples contain a rhetorical or suggestive question. Improved contextualization can again help to identify this kind of comments, when signals such as the existence of question words or exclamation marks are taken into account.

Metaphors and comparisons. Subtle metaphors and comparisons often require understanding of implications of language or additional world knowledge. Zhang and Luo [2019] and Schmidt and Wiegand [2017] also report this common error class.

Example:
"Who are you a sockpuppet for?"

We only see this problem in the Wikipedia dataset samples with 16% of false negatives impacted.

Sarcasm and irony. Nobata et al. [2016] and Qian et al. [2018] report the problem of sarcasm for hate speech detection. As sarcasm and irony detection is a hard task itself, it also increases difficulty of toxic comment classification, because the texts usually state the opposite of what is really meant.

Example:
"hope you're proud of yourself. Another milestone in idiocy."

Sarcasm or irony appears in 11% of Wikipedia dataset samples, but in none of the Twitter dataset samples. This error class requires a model that can learn complex patterns since sarcasm and irony can even be misinterpreted by human readers.

3.6.2 Error Classes of False Positives

Doubtful labels. We find that 53% of false positive samples from the Wikipedia dataset actually fall under the definition of *toxic* in our view, even though they are labeled as non-toxic. Most of them contain strong hateful expressions or spam. We identify 10% of the Twitter dataset samples to have questionable labels.

Example:
"IF YOU LOOK THIS UP UR A DUMB RUSSIAN"

The analysis shows that doubtful labels belong to one of the main reasons for a false classification on the Wikipedia dataset, especially for false positives. The results emphasize the importance of taking labeler agreement into account when creating a dataset to train machine learning models. It also shows the need for clear definitions especially for classes with high variance like toxicity. Besides that, a deficient selection of annotators can amplify such problems as Waseem et al. [2018] point out.

Usage of swear words in false positives. Classifiers often learn that swear words are strong indicators for toxicity in comments. This can be problematic when non-toxic comments contain such terms. Zhang and Luo [2019] describe this problem as dealing with non distinctive features.

Example:
"Oh, I feel like such an asshole now. Sorry, bud."

60% of false positive Wikipedia dataset samples and 77% of Twitter dataset samples contain swear words. In this case, the context is not successfully incorporated into the classification decision. Hence, the classifier overrates signals from the trigger word (a swear word), rather than reacting to signals from the context, in this case, a first person statement addressing the author themself.

Quotations or references. We observe many cases of references to toxic or hateful language in actual non-hateful comments.

Example: *"I deleted the Jews are dumb comment."*

In the Wikipedia dataset samples, this appears in 17% and in the Twitter dataset in 8% of comments. Again, the classifier could not recognize the additional context referring to typical actions within an online community, explicitly expressed with the words "I deleted the ... comment" in the example.

Misspelled and rare words. Rare or idiosyncratic words in non-toxic or non-hateful comments cause problems when the classifier misinterprets their meaning or when they are slang that is often used in toxic language.

Example: *"WTF man. Dan Whyte is Scottish"*

8% of Wikipedia dataset samples include rare words. In the Twitter dataset sample the frequency is higher with 17%, but also influenced by common Twitter language. For this error class, our models would both benefit from additional world knowledge and stronger contextualization.

3.6.3 Summary of Open Challenges

The error analysis on the ensemble of methods revealed open challenges for user comment classification. We summarize them in the following and describe how they could be met by the use of pre-trained large language models.

- **Missing world knowledge** The signals from the training data is oftentimes not sufficient to encode required world knowledge into the model parameters. This hinders the models to correctly interpret more subtle toxicity. Through their cross-domain pre-training and the large number of parameters, LLMs incorporate much more world knowledge than the evaluated baseline models, which could be beneficial for these types of errors.
- **Lack of context incorporation** Although most of the evaluated models have mechanisms to include sequence context into their classification, we observe that this context is not always properly incorporated leading to misclassifications. Transformer-based LLMs use self-attention so that each context token can influence any other token regardless of their distance in the text. Through multiple self-attention layers, the contextuality of the final document representation exceeds those of previous models. In Chapter 4, we analyze these layers in detail.

- **Inconsistent annotations** A major problem in user comment classification and many other specialized domains is the inconsistency of annotations. One reason is a lack of quality of annotations, e.g. when annotations are crowd sourced and the annotators are not properly trained. The second reason is the subjectivity of labels. Whether annotators judge a comment as *toxic* or not depends highly on their personal background. We see similar issues in other specialized domains, such as the clinical, as discussed in 6.6.2. While the training of LLMs also suffers from inconsistent annotations, the large amount of pre-training data increases robustness towards single inconsistent data points due to increased generalization abilities.

3.7 Chapter Summary

In this chapter, we presented multiple neural and statistical approaches for toxic comment classification, presenting the state-of-the-art before pre-trained large language models were introduced. We showed that the approaches make different errors and can be combined into an ensemble with improved performance. The ensemble performs particularly well when there is high variance within the data and on classes with few examples. Some combinations such as shallow learners with deep neural networks are especially effective.

Our error analysis on results of the ensemble identified difficult subtasks of toxic comment classification from which many are transferable to other specialized domains. We find that a large source of errors is the lack of consistent quality of labels. This is especially challenging due to the relatively small amount of available data. Additionally, most of the unsolved challenges occur due to missing context and world knowledge in the evaluated models. This is accompanied with misspelled or rare vocabulary which the models fail to encode into the correct contexts.

Our results show that there is a large potential for the use of pre-trained large language models for toxic comment classification. Recent research confirms this by producing improved results on the task with different LLMs [Zhao et al., 2021b]. Our analysis revealed some of the reasons behind this improvement. Large language models pre-trained on general domain text can complement some of the world knowledge that is missing from the in-domain data. Further, the highly contextualized nature of Transformer-based LLMs leads to improved context incorporation, which caused many errors in previous models. Additional domain-specific pre-training and fine-tuning of LLMs can reduce the influence of inconsistent data points and deliver supplemental signals for rare vocabulary, which is especially useful for specialized domains.

In the next chapter, we investigate in detail how language models are processing and storing the contextual general and task-specific information learned during pre-training and fine-tuning in the parameters of each layer.

4 A Layer-Wise Analysis of Transformer Representations

Large language models based on Transformers are composed of multiple deep network layers that transform inputs into contextualized representations. What these transformations look like and which knowledge is encoded in the learned parameters is, however, an open question, which needs to be investigated in order to understand the possibilities of LLM adaptation. This chapter addresses research question 2: *How do large language models process information throughout their layers?* We approach this question by analyzing models trained on Question Answering, an information-rich task that often requires multiple reasoning steps.

4.1 Introduction

Explainability and model transparency are two important concepts, the absence of which can and should impede the application of neural networks to real-world problems. At the same time, they are difficult to incorporate into the large, black box LLMs that achieve state-of-the-art results in a multitude of NLP tasks. *Bidirectional Encoder Representations from Transformers* (BERT) [Devlin et al., 2019] is one such black box model and the first to display significant improvements over previous state-of-the-art models in a number of different benchmarks and tasks. It has become a staple architecture to solve many different NLP tasks and has inspired a number of related Transformer models. Understanding how these models draw conclusions is crucial for both their improvement and application—not only in specialized domains.

Understanding black box models is also an increasingly prominent area of research [Danilevsky et al., 2020]. While the performance of neural networks has been steadily improving in nearly every domain, our ability to understand how they work, and how they come to the conclusions they draw is only improving slowly. In order for LLMs to be confidently deployed in safety-critical applications, features like transparency,

interpretability and explainability are paramount.

While the inherent attention mechanisms within Transformer models offer an avenue for explainability, Jain and Wallace [2019] argue that attention in fact is not ideal for these purposes, or should at least not be fully relied upon. We take this as motivation to investigate an approach that might add complementary information. Instead of the attention values, we probe and visualize the hidden states after each Transformer encoder layer. This way, we can analyze how the token representations are transformed throughout the network.

The goal of our analysis is to answer the following questions:

1. How are input information processed throughout the layers of Transformer-based LLMs?
2. Are specific layers responsible for encoding different levels of information?
3. What kind of knowledge is encoded in the network during pre-training and during fine-tuning?
4. Can we recognize prediction failure in early layers of the model?

We approach these questions by analyzing pre-trained models that we fine-tune on three Question Answering (QA) datasets. Question Answering is a downstream task that often requires solving multiple other natural language processing tasks step-by-step. These can include Entity Recognition, Coreference Resolution and Relation Extraction. The finding that other NLP tasks can be framed as QA tasks aswell (e.g. shown by McCann et al. [2018]) further extends the possibilities using this setup. Our layer-wise analysis includes a quantitative and a qualitative part. We first apply probing tasks to quantify information encoded in BERT's layers and then use dimensionality reduction to qualitatively study the change in token representations after each layer.

Interactive web demo. To make the findings presented in the following sections more accessible and reproducible, we release VisBERT (https://visbert.demo.datexis.com), an interactive web tool that allows an interpretable visualization of the token representations within BERT-based models trained on the three QA datasets used in this chapter.

4.2 Related Work

There are many different approaches to explaining deep neural models and Transformer-based LLMs, in particular, as surveyed by Lipton [2018], Guidotti et al. [2019], Dosilovic et al. [2018], Rogers et al. [2020] among others. In the following, we focus

on the publications most closely related to our work, using probing and visualizations for this purpose.

Probing Transformer-based LLMs. Probing neural models is a well-established method to gain an understanding of what information is encoded within their parameters. There have been several studies applying probing tasks to such models [Shi et al., 2016, Belinkov et al., 2017, Conneau and Kiela, 2018]. Language models, in particular, have been probed by Tenney et al. [2019], who introduce an edge-probing framework allowing to probe multiple NLP tasks in a unified setup. The framework includes nine different tasks, which are applied to token representations contextualized by ELMo, BERT, and GPT-1. We base our work on their framework, but instead of only studying the pre-trained models, we apply the probing tasks to both fine-tuned and not fine-tuned (but pre-trained) model variants.

Further studies probing BERT-based models are proposed by Goldberg [2019], adding further probing tasks, and Qiao et al. [2019a] focussing on BERT used in a ranking scenario. Similar to Tenney et al. [2019], they analyze pre-trained models only. Liu et al. [2019b] propose a layer-wise analysis of token representations similar to our work. However, the authors do not consider downstream tasks like Question Answering, which allows us to understand the internal phases of solving a complex reasoning task in this work.

Explaining models through visualizations. Another line of research studies the characteristics of deep neural networks qualitatively through visual analysis. Zhang and Zhu [2018] presents an overview of a multitude of methods following this approach. However, their survey is limited to Convolutional Neural Networks. Li et al. [2016] analyze word embeddings and the impact of certain dimensions on the performance regarding sequence tagging and classification tasks. Many approaches using visualizations further focus on visualizing the attention values within Transformer-based models [Vig, 2019]. While this approach can generally lead to insights, Jain and Wallace [2019] show in their work that explanations based on attention values can also be contradictory and sometimes misleading. Their study motivates our work towards the visualization of hidden states instead of attention values.

4.3 Datasets and Models

4.3.1 Datasets

Our aim is to understand how BERT works on complex downstream tasks. Question Answering (QA) is one of such tasks that require a combination of multiple simpler tasks such as Coreference Resolution and Relation Modeling to arrive at the correct answer. We take three current Question Answering datasets into account, namely

	SQuAD	HotpotQA Distr.	HotpotQA SP	bAbI
Baseline	77.2	66.0	66.0	42.0
BERT	87.9	56.8	80.4	93.4
GPT-2	74.9	54.0	64.6	99.9

Table 4.1. Fine-tuning results on the three Question Answering tasks in macro-averaged F1. We compare BERT against GPT-2 and baselines proposed by the task authors: BIDAF [Seo et al., 2017] for SQuAD, an LSTM-based model from [Weston et al., 2016] for bAbI, and the HotpotQA baseline proposed by [Yang et al., 2018].

SQuAD [Rajpurkar et al., 2016], bAbI [Weston et al., 2016] and HotpotQA [Yang et al., 2018]. We intentionally choose three very different datasets to diversify the results of our analysis.

SQuAD. As one of the most popular QA tasks, the SQuAD dataset contains around 100,000 natural question-answer pairs on more than 500 Wikipedia articles. A new version of the dataset called SQuAD 2.0 [Rajpurkar et al., 2018] additionally includes unanswerable questions. We use the previous version SQuAD 1.1 for our experiments to concentrate on the base task of span prediction. The dataset is characterised by questions that mainly require to resolve lexical and syntactic variations.

HotpotQA. This multihop QA task contains about 112,000 natural question-answer pairs. The questions are especially designed to combine information from multiple parts of a context. We focus on the *distractor*-task of HotpotQA, in which the context is composed of both supporting and distracting facts with an average size of 900 words. As the pre-trained BERT model is restricted to an input size of 512 tokens, we reduce the amount of distracting facts by a factor of 2.7 to not exceed the context size. We also add a second task setup which we call *support only* (SP). Here, we only use the supporting facts, i.e. the sentences required to answer the question, as context. This makes the task easier, but also allows to closer analyze the internal processes within the fine-tuned model. In both setups, we leave out yes/no-questions (7% of questions) as they require an additional specific architecture, diluting our analysis.

bAbI. The QA bAbI tasks are a set of artificial toy tasks developed to further understand the abilities of neural models. The 20 tasks require reasoning over multiple sentences (multihop QA) and are modeled to include positional reasoning, argument relation extraction and resolution of coreferences. The tasks strongly differ from the other QA tasks in their simplicity (e.g. vocabulary size of less than 250 and short contexts) and the artificial nature of sentences.

4.3.2 Experimental Setup

Models. For both the probing and the visualization of hidden states, we use the publicly available BERT implementations and models published by Wolf et al. [2020]. More specifically, we use the two English models *bert-base-uncased* and *bert-large-uncased*. For bAbI and SQuAD, the base version of the model reaches state-of-the-art results, while the more complex HotpotQA task requires the larger version of the model. The context lengths of bAbI and SQuAD are also shorter than for HotpotQA, so we fix their input length to 384 tokens, while using 512 tokens for the two HotpotQA task setups corresponding to the maximum input size for the BERT model.

Training. For fine-tuning, we base our setup on the pre-trained models and add randomly initialized classification heads on top of their encoder blocks. For the bAbI dataset, we use a sequence classification head and for the other two datasets, we use span prediction heads, as required by the task specifications. We tune all models regarding the hyperparameters learning rate (+ scheduling) and batch size with a grid search approach. For our analysis, we use the resulting best-performing models. For the bAbI tasks, we evaluate models fine-tuned separately per task and additionally evaluate a multitask model trained on all 20 bAbI tasks jointly.

Probing. Figure 4.1 shows our probing setup. Question and context tokens are processed by N encoder blocks with a positional embedding added beforehand. The output of the last layer is fed into a prediction head consisting of a linear layer and a softmax. We use the hidden states of each layer as input to a set of probing tasks to examine the encoded information.

4.4 Layer-wise Analysis of Hidden States

We want to understand how information is processed internally by BERT's encoder layers. Towards this goal, we use both a quantitative and a qualitative approach. We first apply a probing setup to measure the degree of language-specific information within each layer output for the *quantitative* analysis and then observe the relative position of tokens in vector space after each layer for a *qualitative* analysis of the transformations happening within the model.

4.4.1 Probing BERT's Layers

Our goal is to understand the abilities of the model after each input transformation. We, therefore, apply a set of semantic probing tasks to analyze which information is stored within the transformed tokens after each layer. We want to find out whether

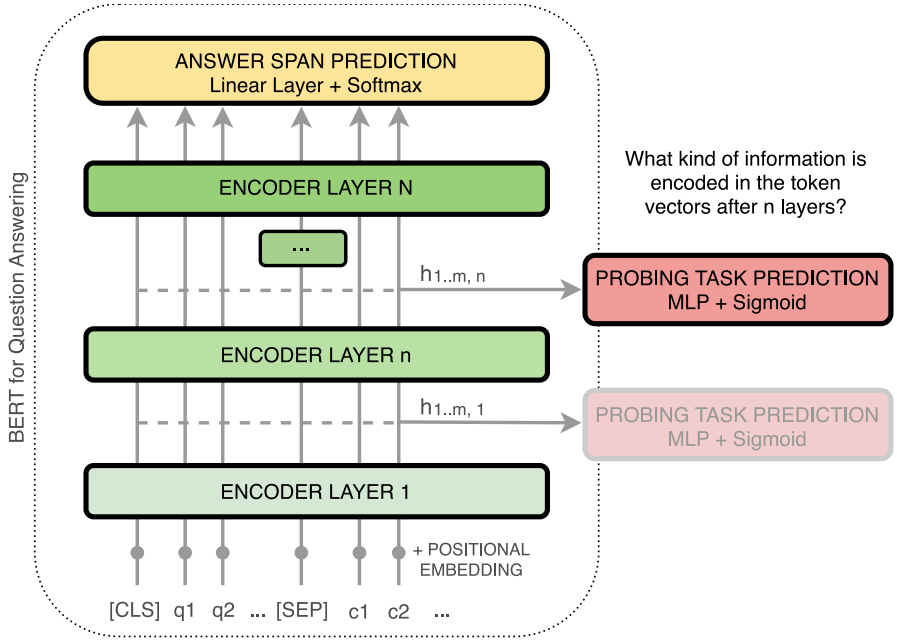


Figure 4.1. Schematic overview of the BERT architecture and our probing setup.

specific layers are *reserved* for specific tasks and how language information is maintained or forgotten by the model.

We use the principle of *Edge Probing* introduced by Tenney et al. [2019]. Edge Probing translates core NLP tasks into classification tasks by focusing solely on their labeling part. This enables a standardized probing mechanism over a wide range of tasks. We adopt the tasks Named Entity Labeling, Coreference Resolution and Relation Classification from the original paper as they are prerequisites for language understanding and reasoning [Weston et al., 2016]. We add the tasks of Question Type Classification and Supporting Fact Identification due to their importance for Question Answering in particular.[1] We introduce the applied probing tasks in the following.

Named Entity Labeling (NEL). Given a span of tokens the model has to predict the correct entity category. This is based on Named Entity Recognition but formulated as a classification problem. The task was modeled by [Tenney et al., 2019], annotations are based on the OntoNotes 5.0 corpus [Weischedel et al., 2011] and contain 18 entity categories.

Coreference Resolution (COREF). The Coreference Resolution task requires the model to predict whether two mentions within a text refer to the same entity.

[1] The source code is available at https://github.com/bvanaken/explain-BERT-QA.

The task was built from the OntoNotes corpus and enhanced with negative samples by [Tenney et al., 2019].

Relation Classification (REL). In Relation Classification, the model has to predict which relation type connects two known entities. The task was constructed by Tenney et al. [2019] with samples taken from the SemEval 2010 Task 8 dataset consisting of English web text and nine directional relation types.

Question Type Classification. A fundamental part of answering a question is to correctly identify its question type. For this Edge Probing task we use the Question Classification dataset constructed by Li and Roth [2002] based on the TREC-10 QA dataset [Voorhees, 2001]. It includes 500 fine-grained types of questions within the larger groups of abbreviation, entity, description, human, location and numeric value. We use the whole question as input to the model with its question type as label.

Supporting Facts Identification. The extraction of supporting facts is essential for Question Answering tasks, especially in the multi-hop case. We examine what BERT's token transformations can tell us about the mechanism behind identifying important context parts. To understand at which stage this distinction is done, we construct a probing task for identifying supporting facts. The model has to predict whether a sentence contains supporting facts regarding a specific question or whether it is irrelevant. Through this task we test the hypothesis that token representations contain information about their significance to the question.

Both HotpotQA and bAbI contain information about sentence-wise supporting facts for each question. SQuAD does not require multi-hop reasoning, we thus consider the sentence containing the answer phrase the supporting fact. We also exclude all QA-pairs that only contain one context sentence. We construct a different probing task for each dataset in order to check their task-specific ability to recognize relevant parts. All samples are labeled sentence-wise with *true* if they are a supporting fact or *false* otherwise.

Probing setup. Analogue to [Tenney et al., 2019], we embed input tokens for each probing task sample with our fine-tuned BERT model. Contrary to previous work, we do this for all layers ($N = 12$ for BERT-base and $N = 24$ for BERT-large), using only the output embedding from n-th layer at step n. The concept of Edge Probing defines that only tokens of labeled edges (e.g. tokens of two related entities for Relation Classification) within a sample are considered for classification. These tokens are first pooled for a fixed-length representation and afterwards fed into a two-layer Multi-layer Perceptron (MLP) classifier, that predicts label-wise probability scores (e.g. for each type of relation). A schematic overview of this setting is shown in Figure 4.1. We perform the same steps on pre-trained BERT-base and BERT-large

models without any fine-tuning. This enables us to identify which abilities the model learns during pre-training and fine-tuning.

4.4.2 Visualization of Transformed Tokens in Vector Space

In Transformer-based networks, each input token is being transformed multiple times into a final contextualized representation. While this representation is strongly affected by the surrounding tokens, it is still relatable to its original input token. This allows us to directly follow and analyze the transformation of single tokens through the network layers by observing how token vectors change after each layer.

In contrast to analyzing the single attention weights within BERT's attention heads, this method allows us to observe the actual outcomes of the whole encoder module in each layer. Since each layer of BERT outputs a different distribution of

	HotpotQA
Question	What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell?
Answer	**United States ambassador**
Context	Kiss and Tell (1945 film): **Kiss and Tell is a 1945 American comedy film starring then 17-year-old Shirley Temple as Corliss Archer.** In the film, two teenage girls cause their respective parents much concern when they start to become interested in boys. The parents' bickering about which girl is the worse influence causes more problems than it solves. Meet Corliss Archer: Meet Corliss Archer, a program from radio's Golden Age, ran from January 7, 1943 to September 30, 1956. Although it was CBS's answer to NBC's popular "A Date with Judy", it was also broadcast by NBC in 1948 as a summer replacement for "The Bob Hope Show". From October 3, 1952 to June 26, 1953, it aired on ABC, finally returning to CBS. Despite the program's long run, fewer than 24 episodes are known to exist. Shirley Temple: Shirley Temple Black (April 23, 1928 - 2013 February 10, 2014) was an American actress, singer, dancer, businesswoman, and diplomat who was Hollywood's number one box-office draw as a child actress from 1935 to 1938. **As an adult, she was named United States ambassador to Ghana and to Czechoslovakia and also served as Chief of Protocol of the United States.** Janet Waldo: Janet Marie Waldo (February 4, 1920 - 2013 June 12, 2016) was an American radio and voice actress. She is best known in animation for voicing Judy Jetson, Nancy in "Shazzan", Penelope Pitstop, and Josie in "Josie and the Pussycats", and on radio as the title character in "Meet Corliss Archer"

Table 4.2. Sample from HotpotQA dataset. Supporting facts are printed in bold. The context includes distracting facts, which contain words that intentionally overlap with the words in the question.

token vectors and we do not have a reference for semantic meanings of positions within these vector spaces, we consider distances between token vectors as indication for semantic relations.

Processing BERT's hidden states For a given input QA sample we collect the hidden states from each layer while removing any padding. We then visualize the input on a token-by-token basis. To that end, we use the hidden states after each Transformer encoder block, which contains a vector for each token with a dimensionality of 768 (BERT-base) or 1024 (BERT-large). Since these high dimensional vectors are not directly interpretable we apply dimensionality reduction, mapping the vectors into a two-dimensional space. We evaluate the dimensionality reduction techniques T-distributed Stochastic Neighbor Embedding (t-SNE) [van der Maaten, 2009], Principal Component Analysis (PCA) [F.R.S., 1901] and Independent Component Analysis (ICA) [Comon, 1994] and find that PCA is most suitable for this scenario as it reveals clusters that correspond to those observed by k-Means clustering [Lloyd, 1982]. We thus use PCA for the qualitative analysis and the VisBERT tool.

The dimensionality reduction result is a 2D representation of each token throughout the model's layers. We further categorize the tokens based on affiliation to question, supporting facts (facts that are necessary to answer the question) or predicted answer in order to facilitate interpretability.

4.5 Results and Discussion

Training results. Table 4.1 shows the evaluation results of our best models. Accuracy on the SQuAD task is close to human performance, indicating that the model can fulfill all sub-tasks required to answer SQuAD's questions. As expected, the tasks derived from HotpotQA prove much more challenging, with the *distractor* setting being the most difficult to solve. Unsurprisingly too, bAbI was easily solved by both BERT and GPT-2. While GPT-2 performs significantly worse in the more difficult tasks of SQuAD and HotpotQA, it does considerably better on bAbi reducing the validation error to nearly 0. Most of BERT's error in the bAbI multi-task setting comes from tasks 17 and 19. Both of these tasks require positional or geometric reasoning, thus, we can assume that this is a skill in which GPT-2 improves on BERT's reasoning capabilities.

Presentation of analysis results. The qualitative analysis of vector transformations reveals a range of recurring patterns. In the following, we present these patterns by a representative sample from the HotpotQA dataset showed in Table 4.2. As explained in 4.3, we split the HotpotQA task into the two separate tasks *distractor* and *support only* (SP). We present results on both tasks. While BERT's performance is

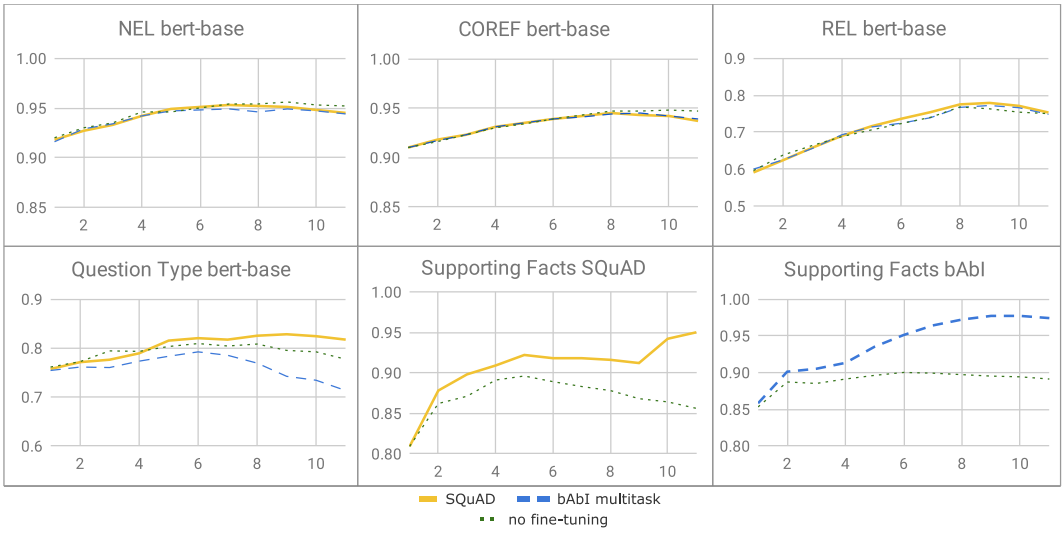


Figure 4.2. Probing task results of BERT-base models in macro averaged F1 (Y-axis) over all layers (X-axis). Fine-tuning barely affects accuracy on NEL, COREF and REL indicating that those tasks are already sufficiently covered by pre-training. Performances on the Question Type task shows its relevancy for solving SQuAD, whereas it is not required for the bAbI tasks and the information is dropped.

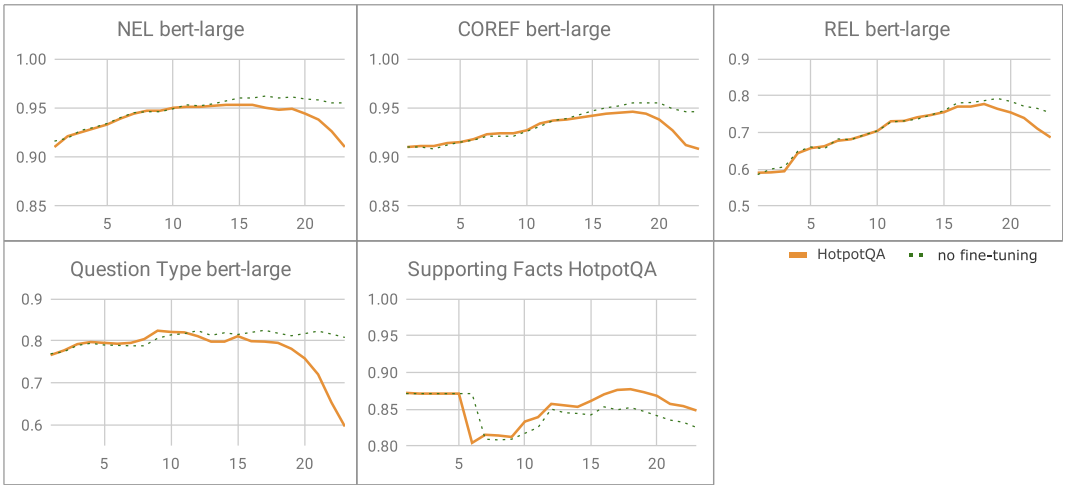
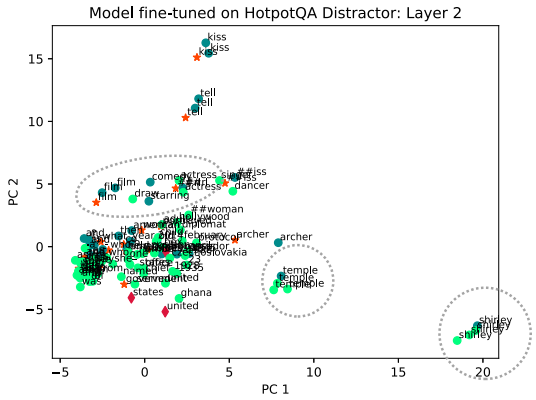


Figure 4.3. Probing task results of BERT-large models in macro averaged F1 (Y-axis) over all layers (X-axis). Performance of the HotpotQA model is mostly equal to the model without fine-tuning, but information is dropped in last layers in order to fit the answer selection task.

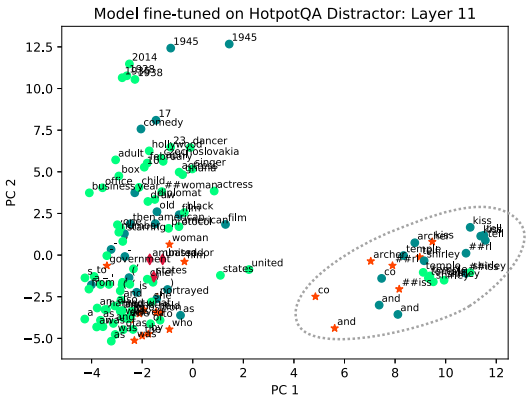
notably better at the SP task, we can observe similar patterns within both vector representations.

Further examples from the SQuAD and bAbI dataset can be found in the interactive VisBERT demo application.

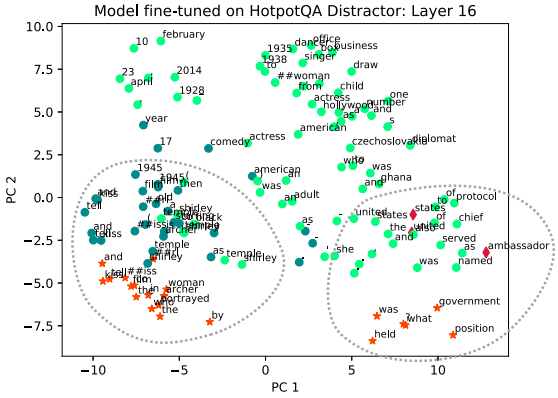
Results from probing tasks are displayed in Figures 4.2 and 4.3. We compare results in macro-averaged F1 over all network layers. Figure 4.2 shows results from three models of BERT-base with twelve layers: Fine-tuned on SQuAD, on bAbI tasks and without fine-tuning. Figure 4.3 reports results of two models based on BERT-large with 24 layers: Fine-tuned on HotpotQA and without fine-tuning.



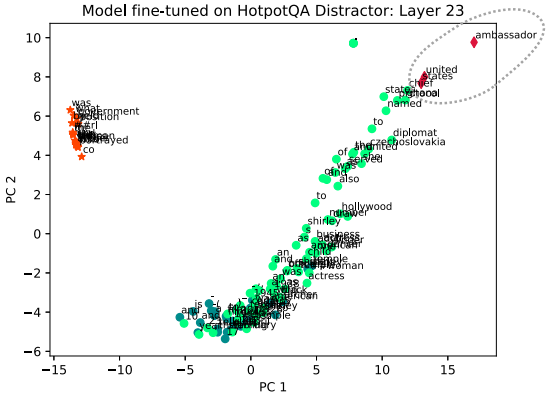
(a) HotpotQA SP phase 1: Topical clustering.



(b) HotpotQA SP phase 2: Connecting entities.



(c) HotpotQA SP phase 3: Matching questions with supporting facts.



(d) HotpotQA SP phase 4: Answer extraction.

Figure 4.4. BERT's transformation phases for the HotpotQA *SP* example from Table 4.2. Answer token: Red diamond-shaped. Question tokens: Orange star-shaped. Supporting fact tokens: Dark cyan and light green. Prominent clusters are circled. Compared to the *distractor* task, we see that there is a clearer separation between clusters due to the lack of distracting tokens.

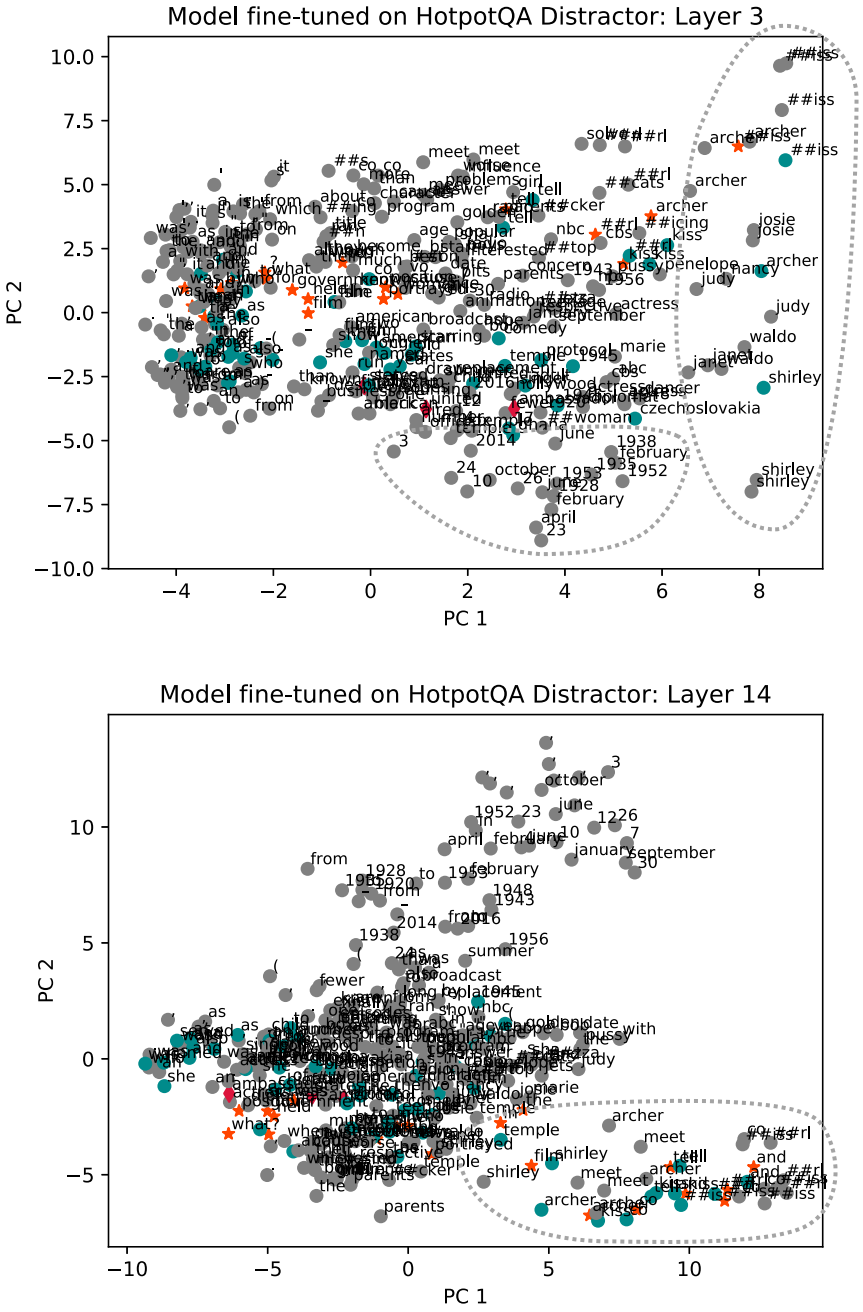


Figure 4.5. HotpotQA *distractor* first and second phase: Topical clustering (top) and connecting entities (bottom).

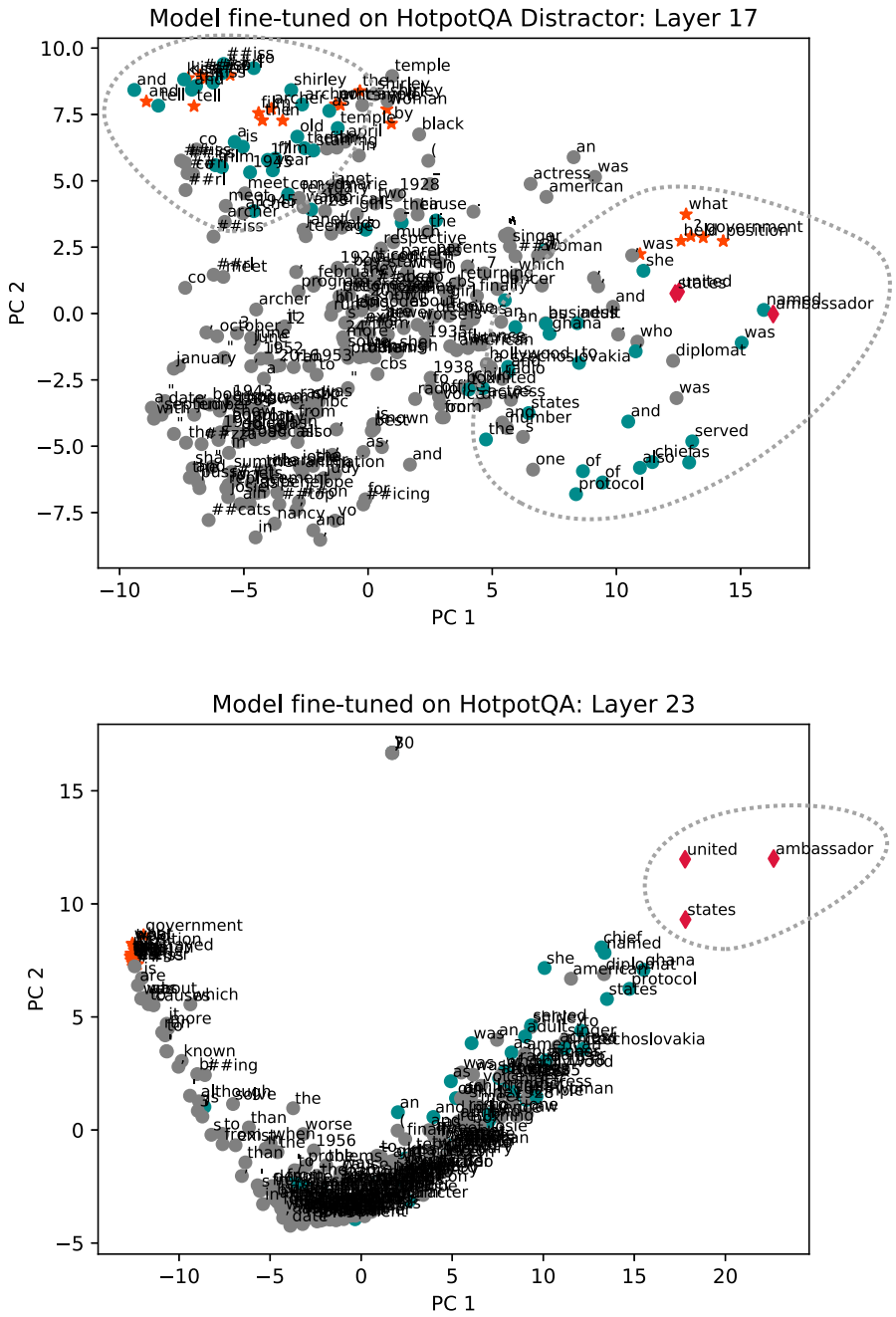


Figure 4.6. HotpotQA *distractor* third and fourth phase: Matching questions with supporting facts (top) and answer extraction (bottom).

4.5.1 Phases of BERT's Inference

The quantitative and qualitative analysis we conducted reveal that BERT models pass multiple phases while answering a question. These phases can be observed throughout the three analyzed QA tasks despite their differences. We describe the observed phases in the following, refering to the results of the probing tasks (Figure 4.2 and Figure 4.3) and the qualitative analysis of vector representations (presented in Figure 4.4 - 4.6).

1. **Topical clustering** In the first layers, we see that tokens are clustered based on topical similarities, comparable to a static word embeddings like Word2Vec [Mikolov et al., 2013]. For example, Figure 4.5 shows tokens clustered by topical groups such as names and dates. The tokens show no task-specific or contextual relations in the first layers. This corresponds to a low F1 score on probing tasks that require semantic and contextual information such as Relation Classification (REL) as shown in Figure 4.2 and 4.3.

2. **Connecting entities with mentions and attributes** Middle layers tend to cluster tokens based on their relation in the specific context. We see multi-token entities clustered together, since their tokens share one semantic meaning. We can also observe clusters of entities with their specific attributes. In Figure 4.5, we see that the entity *Shirley Temple* got matched with *Corliss Archer* and the two programmes she starred in, namely *Kiss and Tell* and *Meet Corliss Archer*. The probing results show similar patterns: Information about named entities (NEL) and their mentions (COREF) are increasingly encoded in the token vectors until the higher layers of the network. In the second phase, complex relations (REL task) are not yet fully encoded in the vectors of both BERT base and BERT large.

3. **Matching questions with supporting facts** In the third quarter of BERT layers, we can see that the question tokens form clusters with the tokens of supporting facts. In multi-hop questions, we even observe clusters for each *hop* that the question contains. Figure 4.6 shows that the question tokens are clustered towards the supporting fact tokens. We see this in the majority of HotpotQA samples, in which the questions are usually build up by two or three individual questions. BERT recognizes this split and can, as we see in phase 4, distinguish the question part that points towards the answer (in this case *Which government position was held...?*). The scores of the probing tasks peak in these layers showing that most semantic and contextual information is stored in the vector representations at this phase.

4. **Answer extraction** In the last layers, the answer tokens are separated from all other tokens. Earlier semantic clusters are dissolved. Based on the certainty of the decision, there might be other potential candidate tokens separated from

the rest as well, with the furthest answer tokens being chosen as final prediction. For instance, in Figure 4.6, we see that the tokens *Chief of Protocol* were also considered answer candidates. In Figure 4.4d we additionally see how the question tokens are extracted from the rest of the context. This shows that the positional embedding is still maintained within the vector representations and that the model has learned that the answer token is never found in the question, and, thus, separates question tokens from answer candidates. The final layers of the network show the highest task specificity and the probing results even indicate that general language information is dropped from the final token representation, e.g. information about named entities and coreferences.

4.5.2 Additional Findings

Failure states. Decision legitimization is an important aspect of neural network explainability. If a network predicts an answer, it is useful to know why, in order to both improve the network and to understand its limits. The visualizations of tokens show signs of wrong predictions not only in the last layers, even early phases can be helpful in analyzing errors. For example, in cases for which a wrong prediction has the same type as the ground truth answer, the problem is often that the wrong supporting fact was selected. This is clearly visible in layers of phase 3, when the question is matched with a wrong fact. For predictions that are completely wrong (not even of the same type as the answer) the phases often degenerate completely. This results in all layers looking either like a mostly homogeneous cloud of tokens or keep in a mainly topical non-contextual clustering as in phase 1. Lastly, the network's general confidence can be estimated by looking at the clusters in each layer. For samples for which BERT is very confident, the clusters and phases are distinct. The lower the confidence, the more blurry and indistinct the clusters become.

Impact of fine-tuning. In Figure 4.2 and 4.3, we compare the probing scores of the vanilla pre-trained BERT models with the BERT models trained on QA tasks. We can see that fine-tuning has a small impact on the general language abilities of the models. By pre-training the models, we already enable them to encode sufficient information about entities, their mentions and relations in the text. This highlights the importance of effective pre-training and explains the advantage of pre-trained LLMs towards models trained from scratch: Less fine-tuning data is needed for general language understanding abilities and for the encoding of world knowledge (such as common relations between tokens), which enables the fine-tuning to only focus on the requirements of the specific task. This can then be achieved by fewer changes in model weights, which leads to more efficient training with improved results on downstream tasks.

4.6 Chapter Summary

This part of the dissertation aimed towards revealing some of the internal processes within Transformer-based LLMs. We found that analyzing how the input tokens are transformed throughout the BERT model by looking at the intermediate hidden states reveal information about general model capabilities and can even help to explain individual predictions.

Our findings further indicate that the benefits of large language models in relation to previous neural network approaches are related to their highly contextual text representation, and the world and language knowledge learned during pre-training. Through qualitative and quantitative analyses, we gained a clearer understanding of how this information is stored within the models. The lower layers encode the base knowledge built up during pre-training, while the upper layers focus on task-specific representations, mostly learned during fine-tuning.

This way, LLMs for specialized domains can benefit from general (language) knowledge encoded in lower layers, while being able to store highly domain-specific knowledge in the weights of upper layers. We will use the findings of this exploration of Transformer-based LLMs in the second part of this dissertation to introduce techniques for domain-specific model adaptation.

Part II

Adaptation to Specialized Domains

5 Language Models for Clinical Assertion Detection

After exploring the abilities and functionality of large language models in the first part of this dissertation, we focus on different techniques and use cases for the adaptation to specialized domains in the following second part. Pre-trained LLMs are typically adapted to domains and downstream tasks by fine-tuning them on domain-specific data (see 2.2.2 for details). However, due to varying requirements in domain-specific use cases, model adaptation further incorporates the adjustment of tasks to align with these requirements. Additionally, the domain knowledge found in fine-tuning data is often not sufficient to learn all required patterns and relations and must be augmented with additional domain data. Finally, adapting language models to needs of domain experts is another requirement not covered by simple LLM fine-tuning, which we will address in this part of the dissertation.

In the following, we present methods for domain adaptation to the clinical domain, since it incorporates the typical characteristics and challenges of specialized domains, as introduced in 1.1.1. In this chapter, we introduce a common clinical information extraction task from discharge summaries. We show how the task can be formulated to be solved by LLMs and present how different pre-training affects performance and transferability. An error analysis on the best performing model shows the limits of generalization due to a tendency to overfit on simple patterns within the data which lacks the required variety.

5.1 Introduction

The clinical information stored within narrative reports is difficult for humans to access for clinical, teaching, or research purposes [Perera et al., 2013]. To provide high-quality patient care, health professionals need to have better and faster access to crucial information in a summarized and interpretable format. In this chapter, we focus on the task of Assertion Detection as a way towards achieving such format via

information extraction. We study English discharge summaries and the classification of clinical information as demonstrated in Figure 5.1.

Given a piece of text, we need to identify two pieces of information – a medical entity and textual cues indicating the presence or absence of that entity. Medical entity extraction has been studied extensively [Lewis et al., 2020b], we thus focus our work on the task of predicting the *present* / *possible* / *absent* class over a medical entity, addressing an important information need of health professionals. This setting is reflected in the dataset released by the 2010 i2b2 Challenge Assertions Task [de Bruijn et al., 2011], on which we base our main evaluation.

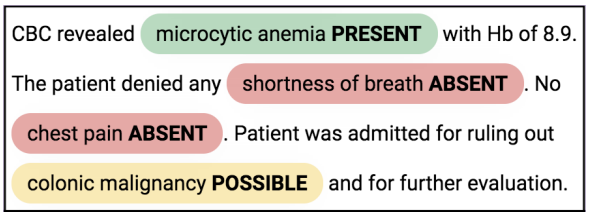


Figure 5.1. Sample output of our demo system. Detected entities are highlighted in red, yellow, and green to indicate *present*, *possible*, and *absent*.

Clinical Assertion Detection is known to be a difficult task [Chen, 2019] due to the free-text format of considered clinical notes. Detecting *possible* assertions is particularly challenging, because they are often vaguely expressed, and they occur far less frequently than *present* and *absent* assertions. Language models pre-trained on medical data have shown to create useful representations for a multitude of tasks in the domain [Peng et al., 2019]. We apply them to our setup of Assertion Detection to evaluate whether they can increase performance (especially on the minority class) and where they still need improvement.

We argue that Clinical Assertion Detection models must be transferable to data that differs from the training data, e.g. due to different writing styles of health professionals from other clinics or from other medical fields. As existing datasets do not represent such diversity, we manually annotate 5,000 assertions in clinical notes from several fields in the publicly available MIMIC-III dataset. We then use these annotated notes as an additional evaluation set to test the transferability of the best performing model.

5.2 Related Work

One of the earliest approaches to Assertion Detection is NegEx [Chapman et al., 2001], where hand-crafted word patterns are used to extract the *absent* category of assertions in discharge summaries. In 2010, the i2b2 Challenge Assertions task [de Bruijn et al., 2011] was introduced, and an accompanying corpus was released.

There is a variety of prior work focused on scope resolution for assertions, which

		present	*possible*	*absent*
2010 i2b2 Challenge Assertion Task	discharge summaries	21,064	1,418	6,144
BioScope	scientific publications	–	3,474	2,161
MIMIC-III Clinical Database	discharge summaries	2,610	250	980
	physician letters	204	34	66
	nurse letters	293	14	59
	radiology reports	249	40	130

Table 5.1. Distribution of text types and classes in the three employed datasets. Note that *possible* is a minority class across datasets as well as text types. In the i2b2 dataset, for instance, only 5% of all labels are *possible*.

differs from our setting in that it does not consider medical concepts but scopes of a certain assertion cue. Representative current approaches for this task setup include a CNN-based (Convolutional Neural Network) one by Qian et al. [2016], reaching an F1 of 0.858 on the more challenging *possible* category. Sergeeva et al. [2019] propose a LSTM-based (Long Short-Term Memory) approach to detect only *absent* scopes. When "gold negation cues" are made available to the model and synthetic features are applied, an F1 of 0.926 is reached. NegBert [Khandelwal and Sawant, 2020] is another approach to detect *absent* scopes. As its name suggests, it is BERT-based and reaches an F1 of 0.957 on BioScope abstracts.

In contrast to these approaches we focus our work on entity-specific Assertion Detection, the results of which are of more practical help for supporting health professionals. Bhatia et al. [2019] explored extracting entities and negations in a joint setting, whereas the work of Harkema et al. [2009], Chen [2019] and de Bruijn et al. [2011] is the closest to our task setup, i.e. labeling entities with an assertion class. Harkema et al. [2009] extended the NexEx algorithm with contextual properties. de Bruijn et al. [2011] use a simple SVM classifier and Chen [2019] apply a bidirectional LSTM model with attention to the task and evaluate it on the i2b2 corpus. While these models reach F1-scores above 0.9 on the majority classes, the challenging *possible* class does not surpass 0.65. We show that medical language models outperform these scores especially regarding the minority class.

Furthermore, Wu et al. [2014] compared then state-of-the-art approaches for negation detection and found a lack of generalization to arbitrary clinical text. We thus want to examine the transfer capabilities of recent language models to understand whether they can mitigate the phenomenon.

5.3 Method

We want to understand the abilities of language models adapted to the clinical domain on the task of Assertion Detection. We hence fine-tune various pre-trained language models on the i2b2 corpus described below. We further apply the best performing model to the BioScope dataset and our newly introduced MIMIC-III assertion dataset without further fine-tuning to test their performance on unseen medical data.

5.3.1 Datasets

The **2010 i2b2 Assertion task** [de Bruijn et al., 2011] provides a corpus of assertions in clinical discharge summaries. The task is split into six classes, namely *present*, *possible*, *absent*, *hypothetical*, *conditional* and *associated with someone else*. However, the distribution is highly skewed, such that only 6% of the assertions belong to the latter three classes. Hence we only use the *present*, *possible*, and *absent* assertions for our evaluation as they present the most important information for doctors.

BioScope [Vincze et al., 2008] is a corpus of assertions in biomedical publications. It was specifically curated for the study of negation and speculation (or *absent* and *possible* in this work) scope and does not contain *present* annotations. As mentioned before, the BioScope dataset does not completely match the information need of health professionals and the i2b2 corpus lacks varied medical text types. We thus introduce a new set of labeled assertions to complement existing data.

The **MIMIC-III Clinical Database** [Johnson et al., 2016] provides texts from discharge summaries as well as other clinical notes (physician letters, nurse letters, and radiology reports) representing a promising source of varied medical text. Therefore, two annotators followed the annotation guidelines from the i2b2 challenge, and labeled 5,000 assertions, i.e. word spans of entities and their corresponding *present* / *possible* / *absent* class. The inter-annotator agreement as Cohen's kappa coefficient is 0.847, which indicates a strong level of agreement. The annotations were further verified by a medical doctor, who provided feedback to correct a small number of labels, and confirmed that the end results were satisfactory. We publish the annotations to encourage future research on Clinical Assertion Detection[1].

It is important to note that even though the newly annotated data from MIMIC-III adds variation to the existing corpora, the dataset has its own limitations. The clinical notes are collected from a single institution (with a mostly White patient population) and from Intensive Care Unit patients only. We therefore argue that progress in assertion detection requires further initiatives for releasing more diverse sets of clinical notes.

Table 5.1 summarizes the assertion distribution in the introduced datasets and shows the unbalanced nature of the data.

[1]The annotations are published at https://github.com/bvanaken/clinical-assertion-data.

5.3.2 Data Preprocessing

We make predictions about assertions on a per-entity level. However, we want our models to consider the context of an entity. We, therefore, pass the whole sentence to the models and surround the entity tokens with special *indicator* tokens `[entity]` whose embeddings are randomly initialized. A sample input sequence thus looks as follows: `[CLS] test results were negative for [entity] COVID-19 [entity]`. We apply the same pre-processing to all three datasets.

5.3.3 Fine-tuning Medical Language Models

There are various pre-trained (bio-)medical and clinical language models available to evaluate on the Assertion Detection task. We select the most prevalent ones and describe them in short below:

BERT [Devlin et al., 2019] was pre-trained on non-medical data and serves as a baseline for Transformer-based pre-trained language models. **BioBERT** [Lee et al., 2020] is a standard model for medical NLP tasks and is pre-trained on bio-medical publications. **Bio+Clinical BERT** and **Bio+Discharge Summary BERT** [Alsentzer et al., 2019] are built upon BioBERT with additional pre-training on clinical notes and discharge summaries respectively. The **CORe** model [van Aken et al., 2021a] uses BioBERT and adds a specialized clinical outcome pre-training as further described in 6.4. **Biomed RoBERTA** [Gururangan et al., 2020] is based on the RoBERTA model [Liu et al., 2019c] and pre-trained on bio-medical publications. After an initial grid search we fix our hyperparameters to a learning rate of 1e-5, batch size of 32, and 2 epochs of training.

Model	F1 for		
	present	*possible*	*absent*
Earlier approaches			
SVM Classifier [de Bruijn et al., 2011]	0.959	0.643	0.939
Conditional Softmax Shared Decoder [Bhatia et al., 2019]	–	–	0.905
Bidirectional LSTM with Attention [Chen, 2019]	0.950	0.637	0.927
Language models under evaluation			
BERT Base [Devlin et al., 2019]	0.968	0.704	0.943
BioBERT Base [Lee et al., 2020]	0.976	0.759	0.963
Bio+Clinical BERT [Alsentzer et al., 2019]	0.977	0.775	0.966
Bio+Discharge Summary BERT [Alsentzer et al., 2019]	**0.979**	**0.786**	**0.972**
Bio+Clinical Outcome Representations (CORe) [van Aken et al., 2021a]	0.975	0.761	0.965
Biomed RoBERTa Base [Gururangan et al., 2020]	0.976	0.723	0.967

Table 5.2. Results of baseline approaches and (medical) language models on the i2b2 Assertions task. Pre-trained medical language models outperform all earlier approaches—with a large margin on the *possible* class. Note that Bhatia et al. [2019] only evaluated their model on negation detection.

	present	*possible*	*absent*
BioScope			
scientific pub.	–	0.593	0.845
MIMIC-III			
discharge sum.	0.951	0.663	0.939
phys. letters	0.929	0.593	0.892
nurse letters	**0.967**	**0.710**	0.900
radio. reports	0.950	0.691	**0.977**

Table 5.3. Experimental results (in F1) for the best performing Bio+Discharge Summary BERT model on two further assertion datasets and their different text types. Both datasets were not seen during training. Note that the number of evaluation samples is very low for some text types (i.e. *possible* class in nurse letters), which impairs the expressiveness of these results.

5.4 Evaluation and Discussion

We start by evaluating the mentioned models on the i2b2 corpus. We use training and test data as defined by in the i2b2 challenge and compare our results to previous state-of-the-art approaches in Table 5.2. Next, we apply the best performing Bio+Discharge Summary BERT to the BioScope and MIMIC-III corpora without additional fine-tuning (Table 8.1). This way we can see the model's performance on medical text from unseen sources.

5.4.1 Results

Language models outperform baselines. Table 5.2 shows that all evaluated medical language models are able to increase F1-scores on all three classes. On the most challenging *possible* class the improvement is the clearest with up to ∼15pp, which shows that the models are better in handling sparse occurrences coupled with vague expressions.

Medical pre-training is important. The vanilla BERT baseline is the weakest of our evaluated models, which shows that models specialized on the medical domain are not only effective for more complex medical tasks but also for Assertion Detection, which is in line with the claim by Gururangan et al. [2020] that domain-specific pre-training is almost always of use. Bio+Discharge Summary BERT is the best model—probably because it was trained on text very similar to the i2b2 corpus.

Text style matters. Table 8.1 shows the ability of the Bio+Discharge Summary BERT language model to transfer to other text styles. The assertions in the BioScope corpus are difficult to identify by the model as they clearly differ from the ones used

by doctors in clinical notes. The text style in MIMIC-III data is more similar to the originally learned data which is reflected in the results.[2] However, physician letters appear to contain more specialized expressions and, therefore, evoke more errors. This points towards a lack of generalization possibly caused by the limited variety of assertion cues in the training data.

5.4.2 Error Analysis

We analyze all errors made by the best performing model Bio+Discharge Summary BERT to identify main sources of errors and to point towards open challenges for further model adaptations.

Inconsistent data. Inconsistent data in pre-existing datasets account for roughly 45% of errors. This includes obvious labeling mistakes, but also disagreements among annotators. For example, phrases such as "appeared to be," "concerning for" and "consistent with" are labeled differently, as *present* or as *possible*.

Long range dependencies. 20% of all errors are found in samples in which entities and their cues have dependencies longer than a few tokens apart. While the model's attention mechanism could easily detect distant tokens, the model might have learned to only consider close assertion cues. The following is an example of a distant cue indicating the *absent* class which was missed by the model:

His rash on the right hand was examined further and is now resolved.

Lists of assertions. 8% of error samples contain lists of assertions. Here the assertion is not directly coupled to an entity but must be inferred by the way it is listed. Such somewhat ambiguous cases are usually easily understood by humans, but difficult for our models.

No hydrocephalus, subarachnoid hemorrhage, no fracture.

Misspellings. Misspelled words account for 5% of all observed errors, but they reveal a critical yet surprising limitation. For instance, the cues "appeas" and "probalbe" that indicate *possible* instances, are missed. While Transformer-based models are generally capable of dealing with misspellings due to subword tokenization, the missing variety of expressions in the data appears to let the models focus on a specific set of textual cues without generalizing to new phrases or even misspellings.

[2]Note that the model's pre-training is based on MIMIC-III and it was thus to an extent exposed to the test data. Due to the difference of the target task and the amount of total pre-training data, this influence should be negligible.

5.5 Chapter Summary

In this chapter, we presented an evaluation on language models to detect assertions in clinical documents. Our experimental results show that language models adapted to the clinical domain via fine-tuning on clinical text outperform baseline approaches. We further provided a new corpus of assertion annotations on the MIMIC-III dataset that will augment existing data collections and shows the model's capability to be transferred to other sources—if the text styles do not strongly differ.

With an error analysis, we identified problems arising from fine-tuning data that lacks variety. We thus see a need to further investigate generalization to unseen data and expressions. Since gathering additional (and more varied) fine-tuning data is not always a feasible option, we examine alternative options for integrating further domain knowledge into the models in the following chapters.

6

CORe: Adapting LLMs to Clinical Outcome Prediction

This chapter addresses a domain-specific information need that is more complex than the information extraction scenario in the last chapter. We examine the task of clinical outcome prediction, which belongs to clinical decision support, and is designed to prevent doctors from overlooking possible risks and help hospitals to plan capacities. In this scenario, we introduce a novel admission to discharge task with four common outcome prediction targets: Diagnoses at discharge, procedures performed, in-hospital mortality and length-of-stay prediction. The ideal system should infer outcomes based on symptoms, pre-conditions and multiple other risk factors of a patient.

We evaluate the effectiveness of large language models to handle this scenario and propose clinical outcome pre-training to integrate knowledge about patient outcomes from multiple public sources. We further present a simple method to incorporate hierarchies of ICD codes, a medical classification system, into the models. This way, we address research question 3: *How can we incorporate domain-specific knowledge into LLMs in the clinical domain?* A detailed analysis further reveals strengths of the model, including transferability, but also weaknesses such as handling of vital values in the text and inconsistencies in the data.

6.1 Introduction

Clinical professionals make decisions about patients under strong time constraints. The patient information at hand is often unstructured, e.g. in the form of clinical notes written by other medical personnel in limited time. Clinical decision support (CDS) systems can help in these scenarios by pointing towards related cases or certain risks. Clinical outcome prediction is a fundamental task of CDS systems, in which the patient's development is predicted based on data from their Electronic Health Record (EHR).

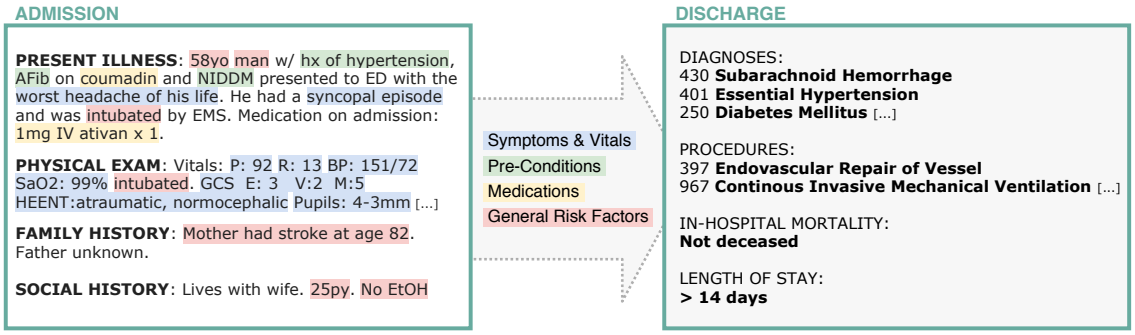


Figure 6.1. *Admission to discharge* sample that demonstrates the outcome prediction task. The model has to extract patient variables and learn complex relations between them in order to predict the clinical outcome.

In this chapter, we focus on textual EHR data available at admission time. Figure 6.1 shows a sample admission note with highlighted parts that – according to medical doctors – must be considered when evaluating a patient.

Encoding clinical notes with pre-trained language models. Neural models need to extract relevant facts from such notes and learn complex relations between them in order to associate certain clinical outcomes. Pre-trained language models such as BERT [Devlin et al., 2019] have shown to be able to both extract information from noisy text and to capture task-specific relations in an end-to-end fashion [Tenney et al., 2019, van Aken et al., 2019]. We thus base our work on these models and pose the following questions:

- Can pre-trained language models learn to predict patient outcomes from their admission information only?
- How can we integrate knowledge about outcomes that doctors gain from medical literature and previous patients?
- How well would these models work in clinical practice? Are they able to interpret common risk factors? Where are they failing?

Simulating patients at admission time. Existing work on text-based outcome prediction focuses on progress notes after a certain time of a patient's hospitalization [Huang et al., 2019]. This is mostly due to a lack of publicly available admission notes and poses some problems: 1) Doctors might miss specific outcome risks early in admission and 2) progress notes already contain information about clinical decisions made on admission time [Boag et al., 2018]. We propose to simulate newly arrived patients by extracting admission notes from MIMIC-III discharge summaries. We are

thus able to give doctors hints towards possible outcomes from the very beginning of an admission and can potentially prevent early mistakes. We can also help hospitals in planning resources by indicating how long a patient might stay hospitalized.

Integrating knowledge with specialized outcome pre-training. Gururangan et al. [2020] recently emphasized the importance of domain- and task-specific pre-training for deep neural models. Consequently, we propose to enhance language models pre-trained on the medical domain with a task-specific *clinical outcome pre-training*. Besides processing clinical language with distinct and specialized terms, our models are thus able to learn about patient trajectories and symptom-disease associations in a self-supervised manner. We derive this knowledge from two main sources: 1) Previously admitted patients and their outcomes. This knowledge is usually stored by hospitals in unlabeled clinical notes and 2) Scientific case reports and knowledge bases that describe diseases, their presentations in patients and prognoses. We introduce a method for incorporating these sources by creating a suitable pre-training objective from publicly available data.

6.2 Related Work

Using clinical notes for outcome prediction. Boag et al. [2018] studied the predictive value of clinical notes with simple approaches such as bag of words. Recent work increasingly applies neural models to compensate for the noisy nature of the data and the complexity of patterns. Hashir and Sawhney [2020] used both convolutional and recurrent layers for outcome prediction, while Jain et al. [2019] and Qiao et al. [2019b] proposed attention-based approaches. Dligach et al. [2019] explored pre-training as a strategy to mitigate data sparsity in clinical setups. Si and Roberts [2019] and Suresh et al. [2018] further showed that outcome prediction benefits from a multitask setup. In contrast to earlier work we apply neural models to admission notes in an *admission to discharge* setup.

Pre-trained language models for the clinical domain. While pre-trained language models are successful in many areas of NLP, there application to the clinical domain has not been studied extensively [Qiu et al., 2020]. Alsentzer et al. [2019] and Huang et al. [2019] both pre-trained BERT-based models on clinical data. They evaluated their work on readmission prediction and other NLP tasks. We are the first to evaluate pre-trained language models on multiple clinical outcome tasks with large label sets. We further propose a novel pre-training objective specifically for the clinical domain.

Prediction of diagnoses and procedures. The majority of work on diagnosis and procedure prediction covers either single diagnoses [Liu et al., 2018, Choi et al.,

2018] or coarse-grained groups [Peng et al., 2020, Sushil et al., 2018]. We argue that models should predict diseases and procedures in a fine-grained manner to be beneficial for doctors. Thus, we use all diagnosis and procedure codes from the data for our outcome prediction tasks.

ICD coding vs. outcome prediction. There is a variety of work in the related field of automated ICD coding [Xie et al., 2018, Falis et al., 2019]. Zhang et al. [2020c] presented a model able to identify up to 2,292 ICD codes from text. However, ICD coding differs from outcome prediction in the way that diseases are directly extracted from text rather than inferred from symptom descriptions and patient history. We further discuss this distinction in Section 6.6.

6.3 Clinical *Admission to Discharge* Task

Clinical outcome prediction can be defined in different ways. We approach the task from a doctor's perspective and predict the outcome of a current admission from the time of the patient's arrival to the hospital unit. We describe our setup as follows.

6.3.1 Clinical Notes from MIMIC-III

As our primary data source, we use the freely-available MIMIC-III v1.4 database [Johnson et al., 2016]. It contains de-identified EHR data including clinical notes in English from the Intensive Care Unit (ICU) of Beth Israel Deaconess Medical Center in Massachusetts between 2001 and 2012. We focus our work on discharge summaries in particular and the outcome information associated with an admission. Similar to previous work, we filter out notes about newborns and remove duplicates.

6.3.2 Creating Admission Notes from Discharge Summaries

The state of a patient is commonly summarized in an ongoing document, which finally concludes in a discharge summary. Since we want to support clinical decisions from the beginning of a patient's stay, we simulate the state of the patient's document at

Admission Notes Statistics			
average (words / doc)	std (words / doc)	average (sentences / doc)	std (sentences / doc)
396.3	233.3	32.5	23.1

Table 6.1. Numbers of words / sentences in MIMIC-III admission notes. We see a high variation in length.

Multi-label tasks: ICD-9 codes per dataset split							
Diagnoses				Procedures			
Total	Train	Val	Test	**Total**	Train	Val	Test
1,266	1,201	906	1,031	**711**	672	476	563

Table 6.2. Distribution of ICD-9 codes per dataset split (patient-wise). Note that very rare codes do not appear in each split of the dataset.

Single-label tasks: Samples per class					
Mortality		Length of Stay (in days)			
0	1	≤ 3	> 3 & ≤ 7	> 7 & ≤ 14	> 14
43,609	5,136	5,596	16,134	13,391	8,488

Table 6.3. Distribution of labels for *Mortality Prediction* and *Length of Stay* task. Both tasks have unbalanced class distributions.

admission time. We thus filter the document by sections that are known at admission such as: *Chief complaint, (History of) Present illness, Medical history, Admission Medications, Allergies, Physical exam, Family history* and *Social history*.

In order to filter the documents by admission sections, we first split all discharge summaries into sections with simple pattern matching. Together with clinical professionals, we then evaluated discharge summaries and identified sections that are known at admission time. We remove all other sections and thus hide information about the further hospital course and discharge of a patient. We exclude notes that do not contain any of the admission sections. We further apply a patient-wise split into train, validation and test set with a 70/10/20 ratio.

Our approach results in 48,745 admission notes. As shown in Table 6.1 the notes contain about 400 words on average. The selection of admission sections as well as the resulting structure of the notes were verified by medical doctors. This newly created admission dataset enables us to make predictions on the outcome of a current admission. At inference time, doctors can then use the model's predictions on textual data from newly arrived patients.

6.3.3 Outcome Prediction Tasks

We select four relevant tasks for outcome prediction in consultation with medical professionals. All tasks take admission notes as input.

Diagnosis prediction. A main goal of clinical outcome prediction is to support medical professionals in the process of differential diagnosis. We thus take all diagnoses associated with an admission into account and frame the task as an extreme multi-label classification. Diagnoses are encoded as ICD-9 codes in the MIMIC-III

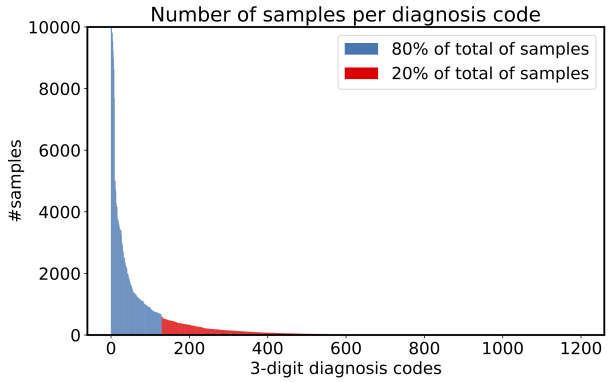


Figure 6.2. Distribution of ICD-9 diagnosis codes in MIMIC-III training set.

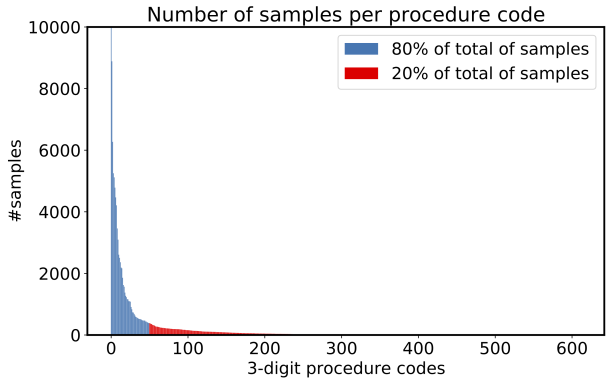


Figure 6.3. Distribution of ICD-9 procedure codes in MIMIC-III training set.

database. Following Choi et al. [2017], we group ICD-9 diagnosis codes from the database from 4- into 3-digit codes to reduce complexity while still obtaining granular suggestions. This results in a total of 1,266 diagnosis codes, which are distributed over our dataset splits as shown in Table 6.2. The labels are power-law distributed with a long tail of very rare codes as shown in Figure 6.2.

Procedure prediction. Procedures are either diagnostics or treatments applied to a patient during a stay. Similarly to diagnosis prediction, this is an extreme multi-label task. We again group the ICD-9 codes from the MIMIC-III database into 3-digit codes. In total, there are 711 procedure codes labeled in the database in a power law distribution similar to the diagnosis codes (Figure 6.3).

In-hospital mortality prediction. Predicting a patient's mortality risk is a fundamental part of the triage process. In-hospital mortality in particular describes whether a patient died during the current admission and is a binary classification task. The percentage of deceased patients in the data is around 10% (see Table 6.3).

As some notes contain direct indications of mortality such as *patient deceased* within the admission sections, we apply an additional filter for those terms.

Length-of-stay prediction. The duration of an ICU stay is an important information for hospitals in order to plan allocations of resources. We group patients into four major categories regarding their length of stay: *Under 3 days, 3 to 7 days, 1 week to 2 weeks, more than 2 weeks.* These categories were recommended by medical doctors in order to make the results as useful as possible in clinical practice. Table 6.3 shows the samples per class.

6.4 Integrating Clinical Knowledge Into Language Models

In the following, we propose *clinical outcome pre-training*, a way to integrate knowledge about clinical patient outcomes into pre-trained language models. We further introduce an additional step to incorporate medical knowledge from the *International Statistical Classification of Diseases and Related Health Problems* (ICD) coding hierarchy into our multi-label classification tasks.

6.4.1 Clinical Outcome Pre-Training

Motivation. Language model pre-training has shown to be of use in specialized domains like the clinical [Alsentzer et al., 2019, Huang et al., 2019]. However, these models lack knowledge about patient trajectories and symptom-diagnosis relations, because their training is focused on learning language characteristics.
We develop an additional pre-training step that produces *Clinical Outcome Representations* (CORe) in order to teach the model relations between symptoms, risk factors and clinical outcomes. Much of this knowledge is present and publicly available, e.g. in knowledge bases like Wikipedia or publication archives like PubMed. Another source is available to hospitals in the form of unlabeled clinical notes from previous patients. The suggested outcome pre-training is a way to use this knowledge to improve the model's capabilities in predicting clinical outcomes as described in 6.3.3.
Corresponding to the way doctors gain their knowledge from both experience and medical literature, we incorporate knowledge from complete patient notes (including discharge information) and medical articles.

Training objective. Our proposed training objective (Figure 6.4) is strongly related to the Next Sentence Prediction (NSP) task introduced by Devlin et al. [2019]. In NSP the model gets two sentences as an input and predicts whether the second follows the first sentence. This way models such as BERT learn relations between

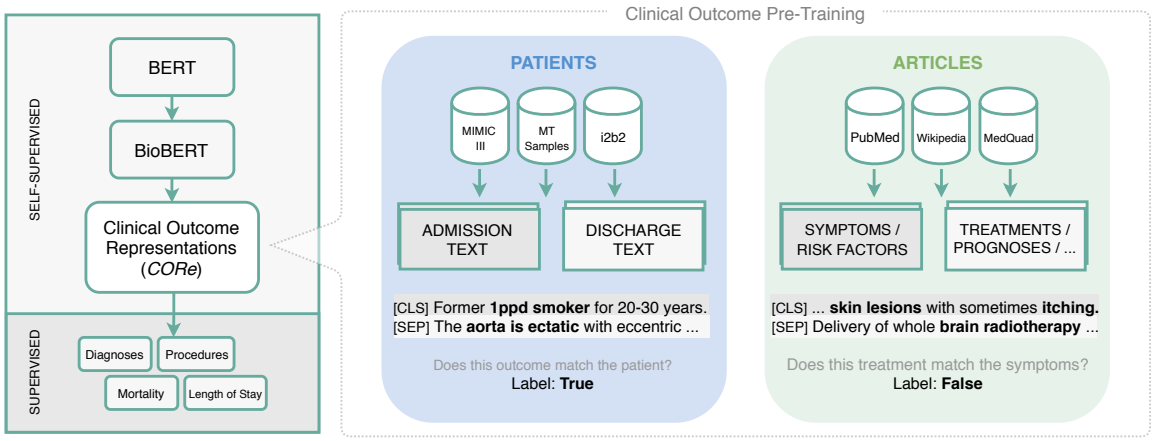


Figure 6.4. Schematic demonstration of *clinical outcome pre-training*. Sources of clinical knowledge are complete patient notes and medical articles. Based on that, we create a self-supervised learning objective that teaches relations between symptoms, risk factors and outcomes.

sentences. We convert this setting so that the model instead learns relations between admissions and outcomes.
From common sections in patient notes, we create two categories: Sections that are created at admission A and sections that are created after admission, e.g. at discharge time D. Given a patient note N, we split it into sections $A_N \in A$ and $D_N \in D$. We remove all other sections. We then sample token sequences from these sections to get $t_{N,1...k} \in A_N$ and $t'_{N,1...k} \in D_N$, where k is randomly set between 30 and 50 tokens. We then train the model to maximize $P(Same\_Patient|X_{N\_N})$ and $P(Other\_Patient|X_{N\_M})$ with

$$\begin{aligned} X_{N\_N} &= Enc(t_{N,1...k}, t'_{\mathbf{N},1...k}) \\ X_{N\_M} &= Enc(t_{N,1...k}, t'_{\mathbf{M},1...k}) \end{aligned} \tag{6.1}$$

with M being a randomly sampled document from the same batch and Enc referring to the BioBERT encoding. As in the original NSP setting, we apply negative sampling (X_{N_M}) for 50% of examples. We apply the same strategy on medical articles and case reports, so that A represents sections describing symptoms and risk factors, and D represents sections that describe outcomes of a disease or case.

Data sources. We create the pre-training dataset from multiple public sources. To integrate knowledge that doctors gain from previous patients and medical literature, we create two groups of sources:

1) *Patients*, which includes 32,721 discharge summaries from the MIMIC-III corpus training set, 5,000 publicly available medical transcriptions from the MTSamples

website [1] and 4,777 clinical notes from the i2b2 challenges 2006-2012[2] [Uzuner et al., 2007, 2008, 2010a,b, 2011, 2012, Sun et al., 2013a,b].

2) *Articles*, composed of 9,335 case reports from PubMed Central (PMC), 2,632 articles from Wikipedia describing diseases and 1,467 article sections from the MedQuAd dataset [Abacha and Demner-Fushman, 2019] extracted from NIH websites such as cancer.gov.

While *Patients* samples contain unaudited practical knowledge, *Articles* samples are built from verified general medical knowledge such as peer-reviewed studies. The sources are, therefore, substantially different and we evaluate their individual effect on performance in Section 6.5.3.

Data preparation. We create admission (A_N) and discharge parts (D_N) of the documents based on section headings. We define common sections belonging to the admission part and those belonging to the discharge part similar to the method described in Section 6.3.2. We ignore sections that cannot be categorized. For section heading extraction from MIMIC-III discharge summaries and MTSamples transcriptions, we apply simple rule-based approaches, which is feasible because the notes are well-structured. For Wikipedia we use headings from the WikiSection dataset [Arnold et al., 2019] filtered for disease articles only. For PubMed Central we similarly use the PubMedSection dataset [Schneider et al., 2020] and filter for section headings that indicate case reports. As i2b2 notes are less well-structured in comparison to MIMIC-III discharge summaries, we use a classifier as proposed by Rosenthal et al. [2019] to determine which section a sentence belongs to. The classifier is trained on an annotated set of i2b2 notes and then applied to all other notes.

6.4.2 ICD+: Incorporation of ICD Hierarchy

Medical knowledge in ICD labels. Diagnosis and procedure prediction requires the model to predict ICD-9 codes in a multi-label manner. ICD-9 codes are hierarchically ordered into associated groups. Figure 6.5 shows the code hierarchy for *Malignant hypertensive renal disease* with the ICD-9 code 403.0. The diagnosis has two parent groups namely *Hypertension renal disease* and *Diseases of the circulatory system*. Diagnoses or procedures in the same group often share similar medical characteristics, therefore, hierarchical relations of a labeled code can be valuable information. This medical information is currently not integrated into the model. The same holds for words describing the ICD-9 codes, that often represent further important signals, such as the words *renal* or *malignant*.

[1] https://mtsamples.com

[2] We exclude notes from the 2014 De-identification and Heart Disease Risk Factors Challenge in order to use this set for evaluation as described in Section 6.5.4.

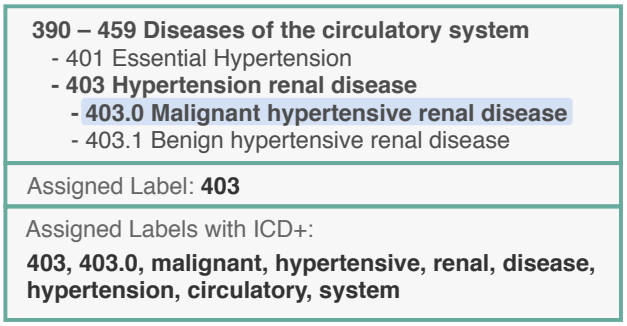


Figure 6.5. Example of ICD+ labeling. *Malignant hypertensive renal disease* is assigned to nine codes (bottom row) that inform about the type and group of the disease.

Enhancing training with useful additional signals. We propose a novel yet simple method, *ICD+*, to incorporate both associated groups and words into the model weights: Instead of only classifying 3-digit codes (as mentioned in 6.3.3), we let the model additionally predict the 4-digit codes and the bag of associated words with a code and its parent groups. In order to create the bag of words per code, we use the descriptions of ICD-9 codes from MIMIC-III and remove all stop words. As shown in Figure 6.5, the ICD+ method assigns eight additional labels to the example diagnosis and thus supplies the model with further information about the diagnosis during training. By increasing the amount of labels per sample, we integrate relevant medical knowledge and enable the model to learn implicit relations between codes and code groups that share certain words. We evaluate the effectiveness of ICD+ in Section 6.5.

6.5 Experimental Evaluation

6.5.1 Training Clinical Outcome Representations

We pre-train the CORe model on top of BioBERT weights[3]. We then fine-tune the model separately on the four outcome tasks. We use the same training regimen for both pre-training and fine-tuning: We tokenize the texts with WordPiece tokenization and truncate them to 512 tokens, due to the limited context length of the pre-trained models. We use early stopping and apply a random search for tuning the following hyperparameters on the validation set: learning rate [1e-4−1e-6], warmup steps [50−30k], dropout [0.1−0.3], class balancing [True/False] (fine-tuning only), gradient accumulation [1−200] with a batch size of 20.

[3]We choose BioBERT as the base for our model because it outperforms BERT on medical tasks and has not seen data from our test set during pre-training unlike DischargeBERT.

6.5.2 Baseline Models

In the following, we introduce the baseline models that we evaluate on the novel outcome prediction tasks. In order to understand the abilities of pre-trained language models we compare their performance against more traditional approaches. The first three models (BOW, word embeddings, CNN) are trained using the hyperparameters proposed by the authors for outcome prediction tasks. The language models are fine-tuned the same way as the CORe model.

Bag of Words. Boag et al. [2018] shows that a simple bag of words (BOW) approach can outperform more complex models on tasks like mortality prediction. We thus include their approach in our evaluation. We adopt their training setting except that we consider 200 instead of 20 top tf-idf words in order to make the model converge.

Pre-trained word embeddings. Boag et al. [2018] further propose the use of pre-computed word embeddings that were trained on MIMIC-III data. We use the same setting as for the BOW approach and fit a support vector machine classifier on the clinical outcome tasks.

Convolutional Neural Network (CNN). Si and Roberts [2019] built a neural network for mortality prediction with two hierarchical convolutional layers at the word and sentence levels and then aggregated it to a patient level representation. We follow their approach to evaluate the model on our four *admission to discharge* tasks.

BioBERT. Following the success of BERT, Lee et al. [2020] further pre-trained the model on biomedical research articles from PubMed using abstracts and full-text articles. They reported improved performance on a range of biomedical text mining tasks.

ClinicalBERT and DischargeBERT. We further evaluate two public language models pre-trained on the clinical domain, with MIMIC-III data in particular. Huang et al. [2019] pre-trained a BERT base model on 100,000 random clinical notes (ClinicalBERT) while Alsentzer et al. [2019] further pre-trained BioBERT on all discharge summaries from MIMIC-III (we refer to the model as DischargeBERT for simplicity).

6.5.3 Results on MIMIC-III Admission Notes

Table 8.1 shows performances in (macro-averaged) area under the receiver operating characteristic curve (AUROC). We report scores of the CORe model trained only on *Articles*, *Patients* and in a combined training setting *CORe All*. We evaluate diagnosis

	Diagnoses (1266 classes)	Procedures (711 classes)	In-Hospital Mortality (2 classes)	Length-of-Stay (4 classes)
BOW [Boag et al., 2018]	75.87	77.47	79.15	65.83
Embeddings [Boag et al., 2018]	75.16	76.72	79.94	66.78
CNN [Si and Roberts, 2019]	61.18	73.13	75.50	64.49
BERT Base [Devlin et al., 2019]	82.08	85.84	81.13	70.40
ClinicalBERT [Huang et al., 2019]	81.99	86.15	82.20	71.14
DischargeBERT [Alsentzer et al., 2019]	*82.86*	*87.09*	***84.51***	*71.73*
BioBERT Base [Lee et al., 2020]	82.81	86.36	82.55	71.59
BioBERT ICD+	83.17	87.45	-	-
CORe Articles (w/o ICD+)	83.46 (82.89)	87.43 (86.75)	83.64	71.99
CORe Patients (w/o ICD+)	83.41 (83.40)	**88.37** (86.60)	83.60	71.96
CORe All (w/o ICD+)	**83.54** (83.39)	87.65 (87.15)	**84.04**	**72.53**

Table 6.4. Results on outcome prediction tasks in macro-averaged % AUROC. The CORe models outperform the baselines, ICD+ adds further improvement (values in parentheses are ablation results without ICD+). DischargeBERT results are printed in italic because the model has seen all test data during pre-training and is thus slightly advantaged.

and procedure prediction both with and without the ICD+ method on BioBERT and the CORe models. In both scenarios we evaluate on 3-digit ICD codes only, in order to maintain comparability between the methods.

Pre-trained models outperform baselines. We see that the evaluated pre-trained language models clearly outperform the BOW, word embeddings and CNN approaches. We further observe that the CORe models improve scores on all tasks in comparison to the baseline models, except for DischargeBERT that reaches a higher score in mortality prediction—probably affected by its exposure to the test data. This shows that even though the language models are trained on similar data (e.g. PubMed and/or clinical notes), the specific *outcome pre-training* improves the model's ability to predict clinical outcome targets. Pre-training on *Patients* and *Articles* achieve similar improvements over the baselines, while the combined training is the most effective. An exception is the procedure prediction, where pre-training on *Patients* achieves the highest score. A probable reason is that procedures are documented in more detail in clinical notes, especially since our selection of medical articles focuses on diseases rather than procedures.

Predicting mortality risk is easier than length of stay. We see that the models reach higher scores in the binary mortality task than in length of stay prediction. Even a simple BOW approach can reach a relatively high score, which indicates that most of the notes contain clear hints towards an increased mortality risk. On the other hand, the length of stay task is difficult due to the many factors that can contribute to the length of a patient's stay after the admission, including nonclinical factors such as the patient's insurance situation [Khosravizadeh et al., 2016].

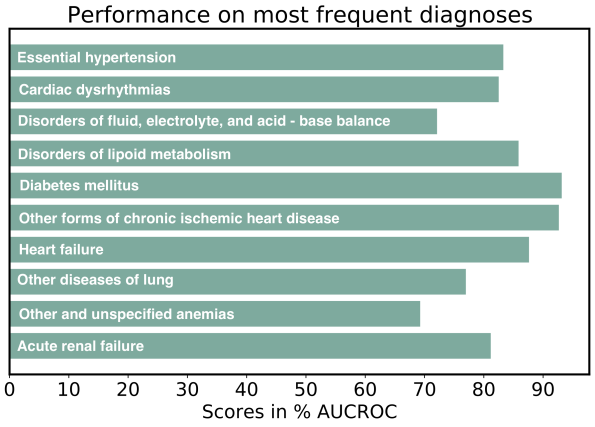


Figure 6.6. Top 10 diagnoses by frequency with the scores reached by the *CORe All* model.

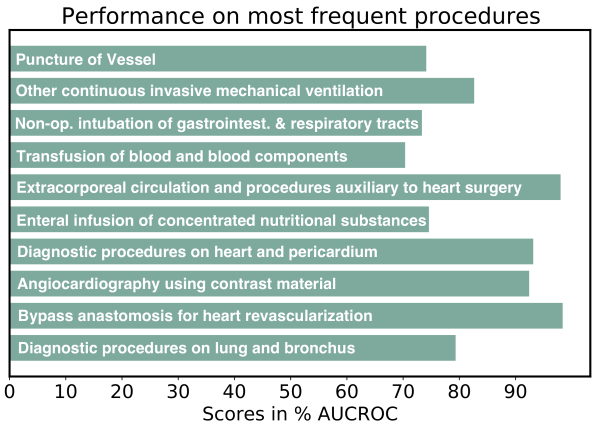


Figure 6.7. Top 10 procedures by frequency with the scores reached by the *CORe All* model.

ICD hierarchy improves diagnosis and procedure predictions. Table 8.1 shows an ablation test without the ICD+ method (in parentheses). We see that both the BioBERT model and the CORe models improve when incorporating code hierarchy and relations through ICD+ into the training process. This is especially visible for ICD procedures, where the hierarchical and textual information, e.g. that a *Nephropexy* is an *operation* on the *kidney* can add important signals during training.

Differing results on most frequent diagnoses and procedures Figures 6.6 and 6.7 show the % AUROC scores of our *CORe All* model on the most frequent labels within the diagnosis and procedure prediction tasks. Figure 6.6 shows that many chronic diseases such as *Essential Hypertension* or *Chronic ischemic heart disease* are among the most common within the MIMIC-III dataset and present with relatively

	i2b2 Diagnoses
BioBERT ICD+	80.43
CORe Articles	81.46
CORe Patients	**82.31**
CORe All	81.15

Table 6.5. Results on i2b2 diagnosis prediction task (5 classes) in % AUROC. The models reach similar results as on the MIMIC-III data, indicating their transferability to other data sources without additional fine-tuning.

high AUROC values. We also observe that very specific codes such as *Diabetes mellitus* and *Bypass Anastomosis* are predicted more easily compared to more general codes such as *Other and unspecified anemias*. Figure 6.7 further shows the negative influence of inconsistent labeling on standard procedures such as *Puncture of Vessel*.

6.5.4 Model Transferability: Cross-Verification on i2b2 Clinical Notes

In order to verify that the fine-tuned models are transferable to ICU data from other sources, we apply it to data from the i2b2 De-identification and Heart Disease Risk Factors Challenge [Stubbs et al., 2015, Stubbs and Uzuner, 2015]. The challenge introduces a dataset that contains clinical notes and discharge summaries annotated based on risk factors and disease indicators. We convert the data into an *admission to discharge* task by selecting five of the annotated conditions which correspond to ICD-9 codes as our labels, namely *Hypertension* (401), *Hyperlipidemia* (272), *Coronary artery disease* (414), *Diabetes mellitus* (250) and *Obesity* (278). Just like the MIMIC-III diagnosis task, samples are annotated in a multi-label fashion. In order to convert the clinical notes to admission notes, we use the dataset from Rosenthal et al. [2019] that contain section labels per sentence. We then exclude sections that are not known at admission time concurrent to Section 6.3.2. This approach results in 1,118 samples labeled with up to five ICD-9 codes.

Models generalize to i2b2 data. We apply the models based on MIMIC-III to predict diagnosis codes for the i2b2 notes without further fine-tuning. We then evaluate whether the predictions contain the five mentioned ICD-9 codes. The results in macro-averaged % AUROC are shown in Table 6.5. Even though the clinical notes differ from the MIMIC-III notes in structure and writing style, the tested models are mostly able to identify the conditions. The scores are comparable to the MIMIC-III results, which shows that the models are able to generalize on data from different sources such as other hospitals.

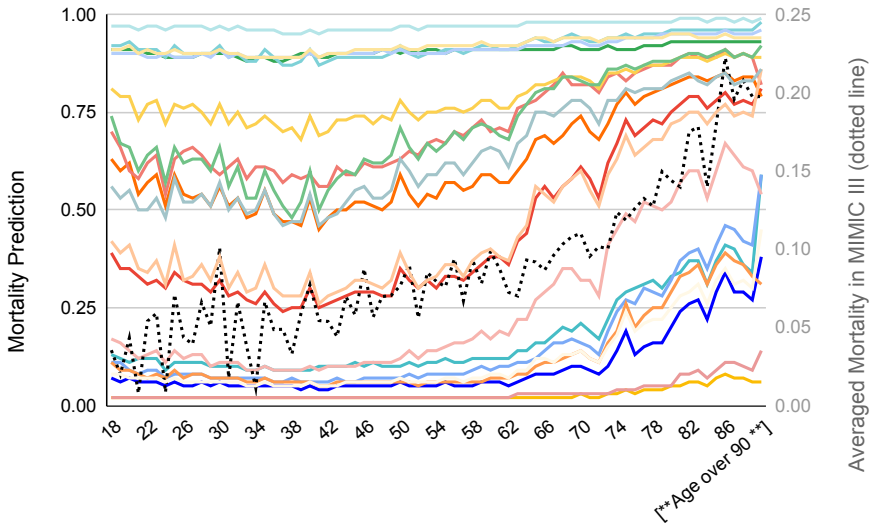


Figure 6.8. Impact of age on mortality prediction on 20 random samples. Mortality risk and age mostly increase proportionally as intended, with certain peaks that might indicate unintended biases in the data.

	% AUROC
All Diagnoses	83.54
Diagnoses **Mentioned** in Text	**87.10**
Diagnoses **Not Mentioned** in Text	82.35

Table 6.6. Analysis of the impact of directly mentioned diagnoses on the diagnosis prediction task. Mentioned diagnoses are detected more reliably. Though on unmentioned diagnoses, scores only see a small decrease compared to the overall score.

6.6 Discussion and Findings

Clinical outcome prediction is a sensitive task. Therefore, we conduct an extensive analysis on the *CORe All* model including a manual error analysis by medical doctors to understand how the model would perform in clinical practice.

6.6.1 A Closer Look at the Model's Abilities

Does the model mainly extract already present diagnoses? We observe that a majority of coded diseases are already mentioned in the admission text. This is mainly due to chronic diseases (e.g. *diabetes mellitus*) or to conditions that were identified prior to the ICU admission (e.g. in the emergency ward). We want to know if our model is also able to predict diagnoses that are not mentioned in the text. We annotate the admission texts with ICD-9 diagnosis codes with the methodology

described by Searle et al. [2020]. We then evaluate on codes that were explicitly mentioned in the text and those that were not. Table 6.6 shows that the model indeed extracts many diagnoses directly from the text and thus reaches a higher score on mentioned diagnoses. On the other hand, we see that the performance on non-mentioned diagnoses does drop only slightly, indicating that the model has also learned to predict non-mentioned diagnoses.

How does age and gender impact predictions? Age and gender are common risk factors with significant impact on the potential clinical outcome of a patient. We want our models to learn that impact without overestimating it. We test the model's behaviour by switching age and gender throughout 20 random samples and analyze how the mortality prediction changes. For each sample we manually switch the age mention and iterate over it from 18 until [**Age over 90**][4]. Figure 7.6 shows that the analyzed samples show a high variation in mortality risk and that age only impacts the prediction partially. In all cases the prediction increases with age—as expected from a medical perspective. We also observe some peaks without a medical reason that are caused by the mortality of certain age groups in the original data (black dotted line). This demonstrates how the model does not follow medical reasoning but merely statistic observations. We similarly switch the gender mention and all pronouns in the texts and observe that mortality prediction for male patients is increased by 5% on average, consistent with medical rationale.

Where is the model failing? To better understand the shortcomings of our model, we present medical professionals with 20 randomly selected error samples.

1. **Negation** Our error analysis finds that negation does not generally falsify the model's predictions, however, we could identify single samples in which especially medical-specific negations, such as *abstinent from alcohol*, are misinterpreted by the model, e.g. into *alcohol dependence syndrome*.

2. **Numerical data** Wallace et al. [2019] show BERT's inabilities to interpret numbers. We observe this in the case that the model does not interpret life-threatening vital values (such as temperature over 105°F) as an increased mortality risk. Clinical notes contain many such relevant values, thus improving the encoding of such data is an important goal for future work.

6.6.2 There is No Ground Truth in Clinical Data

Incomplete and inconsistent labels. Our error analysis reveals that 60% of the analyzed samples are partially under-coded. They contain indicators for a diagnosis or procedure but miss the corresponding ICD-9 code. This is consistent with results from

[4]De-identified age information in MIMIC-III for patients older than 89.

Searle et al. [2020] showing that MIMIC-III is up to 35% under-coded. Additionally we find that procedures that are almost always performed in the ICU such as *Puncture of vessel* are often coded inconsistently. While a doctor can infer these labels with medical common sense, they pose a challenge to our models. We thus suggest a critical view towards the data and welcome additional clinical datasets to compensate for noisy labels.

Multiple possible outcomes. 85% of analyzed samples contain false positive predictions that the doctors still consider medically reasonable. This demonstrates that there are many possible clinical pathways and that some might not be foreseeable at admission time. We also see many cases in which the information in the clinical note is not sufficient and allows multiple interpretations. For future work, we propose including further EHR data as suggested by Khadanga et al. [2019] to extend the patient representation in these scenarios.

6.7 Chapter Summary

In this chapter, we reframed the task of clinical outcome prediction to consider the admission state of a patient and thus support doctors in their initial decision process. We show that large language models outperform selected baselines on this task, and we present methods for further improving them by integrating domain-specific data in a self-supervised manner: First, by *outcome pre-training*, which enables our models to learn from unlabeled sources including clinical case studies and publications. And second, by introducing *ICD+*, a method that incorporates hierarchical and textual ICD representations into our models. Both approaches increase performance and transferability to other datasets.

We further conducted an error analysis and studied the impact of age on mortality predictions. The results of this analysis reveal the need for closer observations of model behavior to ensure safe deployment in real scenarios. That is why we introduce a more comprehensive study on model patterns and behavior in the next chapter.

7 Behavioral Testing of Clinical Language Models

In the previous chapter, we have shown that decision support systems based on clinical notes have the potential to improve patient care by pointing doctors towards overseen risks. Predicting a patient's outcome is an essential part of such systems, for which the use of LLMs has shown promising results.

However, the patterns learned by these networks are mostly opaque and at risk of reproducing unintended biases. We thus introduce an extendable testing framework that evaluates the behavior of clinical outcome models regarding changes in the input. The framework helps to understand learned patterns and their influence on model decisions. In this chapter, we apply it to analyze the change in behavior with regard to the patient characteristics *gender*, *age* and *ethnicity*. We show that communicating model behavior to medical professionals is crucial for the safe application of such systems and that the presented framework can be used as part of such communication. In conjunction with the interpretable model architecture presented in Chapter 8, the following work approaches research question 4: *How can we make large language models more transparent to serve domain requirements?*

7.1 Introduction

The use of automatic systems in the medical domain is promising due to their potential exposure to large amounts of data from earlier patients. This data can include information that helps doctors make better decisions regarding diagnoses and treatments of a patient at hand. Outcome prediction models take patient information as input and then output probabilities for all considered outcomes (see [Choi et al., 2018, Khadanga et al., 2019] and Chapter 6). As in the last chapter, we focus on outcome models using natural language in the form of clinical notes as an input, since they are a common source of patient information and contain a multitude of possible variables.

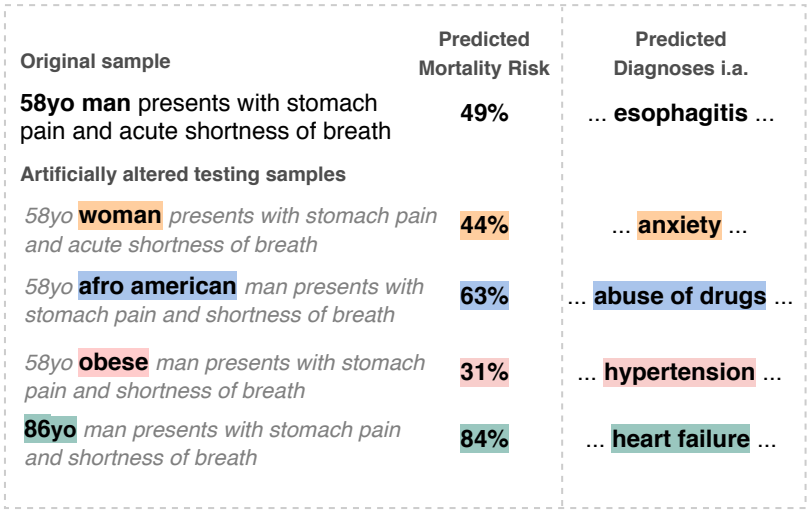


Figure 7.1. Minimal alterations to the patient description can have a large impact on outcome predictions of clinical NLP models. We introduce behavioral testing for the clinical domain to expose these impacts.

The problem of black box models for clinical predictions. Neural models show promising results on tasks such as mortality [Si and Roberts, 2019] and diagnosis prediction [Liu et al., 2018, Choi et al., 2018]. However, since most of these models work as black boxes, it is unclear which features they consider important and how they interpret certain patient characteristics. From earlier work we know that highly parameterized models are prone to emphasize systemic biases in the data [Sun et al., 2019b]. Further, these models have high potential to disadvantage minority groups as their behavior towards out-of-distribution samples is often unpredictable. This behavior is especially dangerous in the clinical domain, since it can lead to underdiagnosis or inappropriate treatment [Straw, 2020]. Thus, understanding models and allocative harms they might cause [Barocas et al., 2017] is an essential prerequisite for their application in clinical practice. We argue that more in-depth evaluations are needed to know whether models have learned medically meaningful patterns or not.

Behavioral testing for the clinical domain. As a step towards this goal, we introduce a novel testing framework specifically for the clinical domain that enables us to examine the influence of certain patient characteristics on the model predictions. Our work is motivated by behavioral testing frameworks for general NLP tasks [Ribeiro et al., 2020] in which model behavior is observed under changing input data. Our framework incorporates a number of test cases and is further extendable to the needs of individual data sets and clinical tasks.

Influence of patient characteristics. As an initial case study we apply the framework to analyze the behavior of models trained on the widely used MIMIC-III

database [Johnson et al., 2016]. We analyze how sensitive these models are towards textual indicators of patient characteristics, such as *age*, *gender* and *ethnicity*, in English clinical notes. These characteristics are known to be affected by discrimination in health care [Stangl et al., 2019], however, they can also represent important risk factors for certain diseases or conditions. That is why we consider it especially important to understand how these mentions affect model decisions.

7.2 Related Work

7.2.1 Clinical Outcome Prediction

Outcome prediction from clinical text has been studied regarding a variety of outcomes. The most prevalent being in-hospital mortality [Ghassemi et al., 2014, Jo et al., 2017, Suresh et al., 2018, Si and Roberts, 2019], diagnosis prediction [Tao et al., 2019, Liu et al., 2018, 2019a] and phenotyping [Liu et al., 2019a, Jain et al., 2019, Oleynik et al., 2019, Pfaff et al., 2020]. In recent years, most approaches are based on deep neural networks due to their ability to outperform earlier methods in most settings. Transformer-based LLMs have been applied for prediction of patient outcomes with reported increases in performance [Huang et al., 2019, Zhang et al., 2020a, Blinov et al., 2020, Zhao et al., 2021a, van Aken et al., 2021a, Rasmy et al., 2021]. In this work, we analyze three of these Transformer-based LLMs due to their upcoming prevalence in the application of NLP in health care.

7.2.2 Behavioral Testing in NLP

Ribeiro et al. [2020] identify shortcomings of common model evaluation on held-out datasets, such as the occurrence of the same biases in both training and test set and the lack of broad testing scenarios in the held-out set. To mitigate these problems, they introduce CheckList, a behavioral testing framework for general NLP abilities. In particular, they highlight that such frameworks evaluate input-output behavior without any knowledge of internal structures of a system [Beizer, 1995]. Building upon CheckList, Röttger et al. [2021] introduce a behavioral testing suite for the domain of hate speech detection to address the individual challenges of the task. Following their work, we create a behavioral testing framework for the domain of clinical outcome prediction, that comprises domain-specific language and data points with respective challenges.

7.2.3 Analyzing Clinical NLP Models

Zhang et al. [2020b] highlight the reproduction of systemic biases in clinical NLP models. They quantify such biases with the recall gap among patient groups and

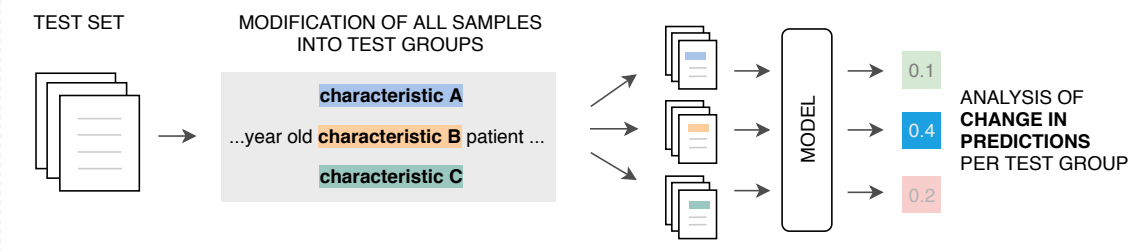


Figure 7.2. Schematic overview of the introduced behavioral testing framework for the clinical domain. From an existing test set we create test groups by altering specific tokens in the clinical note. We then analyze the change in predictions which reveals the impact of the mention on the clinical NLP model.

show that models trained on data from MIMIC-III inherit biases regarding gender, ethnicity, and insurance status—leading to higher recall values for majority groups. Logé et al. [2021] further find disparities in pain treatment suggestions by language models for different races and genders. We take these findings as motivation to directly analyze the sensitivity of large pre-trained models with regard to patient characteristics. In contrast to earlier work and following Ribeiro et al. [2020], we want to eliminate the influence of existing data labels on our evaluation. Further, our approach simulates patient cases that are similar to real-life occurrences. It thus displays the actual impact of learned patterns on all analyzed patient groups.

7.3 Behavioral Testing of Clinical NLP Models

Sample alterations. Our goal is to examine how clinical NLP models react to mentions of certain patient characteristics in text. Comparable to earlier approaches to behavioral testing we use sample alterations to artificially create different test groups. In our case, a test group is defined by one manifestation of a patient characteristic, such as *female* as the patient's gender. To ensure that we only measure the influence of this certain characteristic, we keep the rest of the patient case unchanged and apply the alterations to all samples in our test dataset. Depending on the original sample, the operations to create a certain test group thus include 1) changing a mention, 2) adding a mention or 3) keeping a mention unchanged (in case of a patient case that is already part of the test group at hand). This results in one newly created dataset per test group, all based on the same patient cases and only different in the patient characteristic under investigation.

Prediction analysis. After creating the test groups, we collect the models' predictions for all cases in each test group. Different from earlier approaches to behavioral testing we do not test whether predictions on the altered samples are true or false with regard to the ground truth. As discussed in Section 6.6.2, clinical ground truth must

be viewed critically, because the collected data does only show one possible pathway for a patient out of many. Further, existing biases in treatments and diagnoses are likely included in our testing data potentially leading to meaningless results. To prevent that, we instead focus on detecting how the model outputs change regardless of the original annotations. This way we can also evaluate very rare mentions (e.g. *transgender*) and observe their impact on the model predictions reliably. Figure 7.2 shows a schematic overview of the functioning of the framework.

Extensibility. In this study, we use the introduced framework to analyze model behavior with regard to patient characteristics as described in 7.4.2. However, it can also be used to test other model behavior like the ability to detect diagnoses when certain indicators are present in the text or the influence of stigmatizing language (cf. Goddu et al. [2018]). It is further possible to combine certain patient groups to test model behavior regarding intersectionality. While such analyses are beyond the scope of this work, we include them in the published codebase as an example for further extensions.

7.4 Case Study: Patient Characteristics

7.4.1 Data

We conduct our analysis on data from the MIMIC-III database [Johnson et al., 2016]. In particular, we use the outcome prediction task setup introduced in Chapter 6. The classification task includes 48,745 English admission notes annotated with the patients' clinical outcomes at discharge. We select the outcomes *diagnoses at discharge* and *in-hospital mortality* for this analysis, since they have the highest impact on patient care and present a high potential to disadvantage certain patient groups. We use three models (see 7.4.3) trained on the two *admission* to *discharge* tasks and conduct our analysis on the test set defined by the authors with 9,829 samples.

7.4.2 Considered Patient Characteristics

We choose three characteristics for the analysis in this work: *Age*, *gender* and *ethnicity*. While these characteristics differ in their importance as clinical risk factors, all of them are known to be subject to biases and stigmas in health care [Stangl et al., 2019]. Therefore, we want to test whether the analyzed models have learned medically plausible patterns or ones that might be harmful to certain patient groups. We deliberately also include groups that occur very rarely in the original dataset. We want to understand the impact of imbalanced input data especially on minority groups, since they are already disadvantaged by the health care system [Riley, 2012, Bulatao and Anderson, 2004].

	PubMedBERT	CORe	BioBERT
Diagnoses	**83.75**	83.54	82.81
Mortality	**84.28**	84.04	82.55

Table 7.1. Performance of three state-of-the-art models on the tasks *diagnoses* (multi-label) and *mortality prediction* (binary task) in % AUROC. PubMedBERT outperforms the other models in both tasks by a small margin.

When altering the samples in our test set, we utilize the fact that patients are described in a mostly consistent way in clinical notes. We collect all mention variations from the training set used to describe the different patient characteristics and alter the samples accordingly in an automated setup. Details regarding all applied variations can be found in the public repository[1].

Age. The age of a patient is a significant risk factor for a number of clinical outcomes. Our test includes all ages between 18 and 89 and the [** Age over 90**] de-idenfitication label from the MIMIC-III database. By analyzing the model behavior for changing age mentions we can get insights on how the models interpret numbers, a subtask that is considered challenging for current NLP models [Wallace et al., 2019].

Gender. A patient's gender is both a risk factor for certain diseases and also subject to unintended biases in healthcare. We test the model's behavior regarding gender by altering the gender mention and by changing all pronouns in the clinical note. In addition to *female* and *male*, we also consider *transgender* as a gender test group in our study. This group is extremely rare in clinical datasets like MIMIC-III, but since approximately 1.4 million people in the U.S. identify as transgender [Flores et al., 2016], it is important to understand how model predictions change when the characteristic is present in a clinical note.

Ethnicity. The ethnicity of a patient is only occasionally mentioned in clinical notes and its role in medical decision-making is controversial, since it can lead to disadvantages in patient care [Anderson et al., 2001, Snipes et al., 2011]. Earlier studies have also shown that ethnicity in clinical notes is often incorrectly assigned [Moscou et al., 2003]. We want to know how clinical NLP models interpret the mention of ethnicity in a clinical note and whether their behavior can cause unfair treatment. We choose *White*, *African American*, *Hispanic* and *Asian* as ethnicity groups for our evaluation, as they are the most frequent ethnicities in MIMIC-III.

[1]URL to public repository: https://github.com/bvanaken/clinical-behavioral-testing.

7.4.3 Clinical NLP Models

In this study, we apply the introduced testing framework to three existing clinical models which are fine-tuned on the tasks of diagnosis and mortality prediction. We use public pre-trained model checkpoints and fine-tune all models on the same training data with the same hyperparameter setup[2]. The models are based on the BERT architecture [Devlin et al., 2019] as it presented the state-of-the-art in predicting patient outcomes at the time of these experiments. Their performance on the two tasks is shown in Table 7.1. We deliberately choose three models based on the same architecture to investigate the impact of pre-training data while keeping architectural considerations aside. In general, the proposed testing framework is model agnostic and works with any type of text-based outcome prediction model.

BioBERT. Lee et al. [2020] introduced BioBERT which is based on a pre-trained BERT base [Devlin et al., 2019] checkpoint. They applied another language model fine-tuning step using biomedical articles from PubMed abstracts and full-text articles. BioBERT has shown improved performance on both medical and clinical downstream tasks.

CORe. Clinical Outcome Representations (CORe) introduced in Chapter 6 are based on BioBERT and extended with a pre-training step that focuses on the prediction of patient outcomes. The pre-training data includes clinical notes, Wikipedia articles and case studies from PubMed. The tokenization is similar to the BioBERT model.

PubMedBERT. Gu et al. [2022] introduced the PubMedBERT model based on similar data as BioBERT. They use PubMed articles and abstracts but instead of extending a BERT base model, they train PubMedBERT from scratch. The tokenization is adjusted to the medical domain accordingly. The model reaches state-of-the-art results on multiple medical NLP tasks and outperforms the other analyzed models on the outcome prediction tasks.

7.5 Results

We present the results on all test cases by averaging the probabilities that a model assigns to each test sample. We then compare the averaged probabilities across test cases to identify which characteristics have a large impact on the model's prediction over the whole test set. The values per diagnosis in the heatmaps shown in Figure 7.3, 7.4, 7.7 and 7.8 are defined using the following formula:

[2]Batch size: 20; learning rate: 5e-05; dropout: 0.1; early stopping patience: 20; warmup steps: 1000.

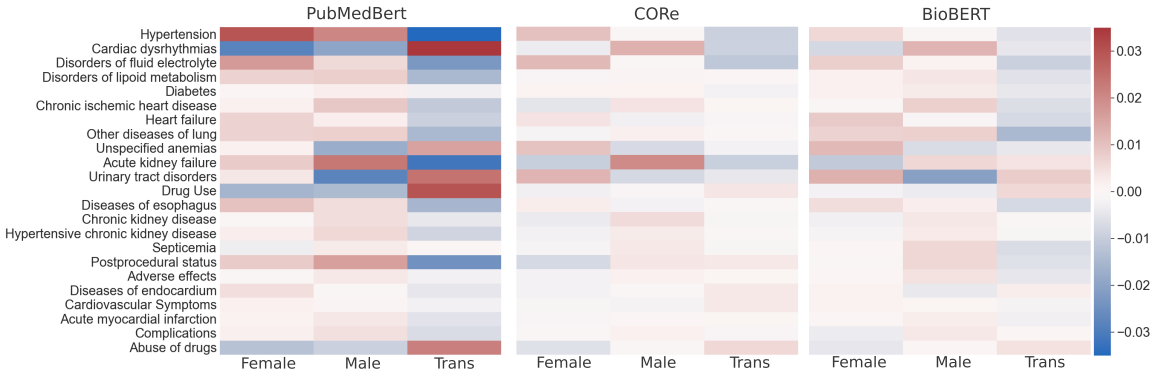


Figure 7.3. Influence of **gender** on predicted diagnoses. Blue: Predicted probability for diagnosis is below-average; red: predicted probability above-average. PubMedBERT shows highest sensitivity to gender mention and regards many diagnoses less likely if *transgender* is mentioned in the text. The graph shows deviation of probabilities on 24 most common diagnoses in the test set.

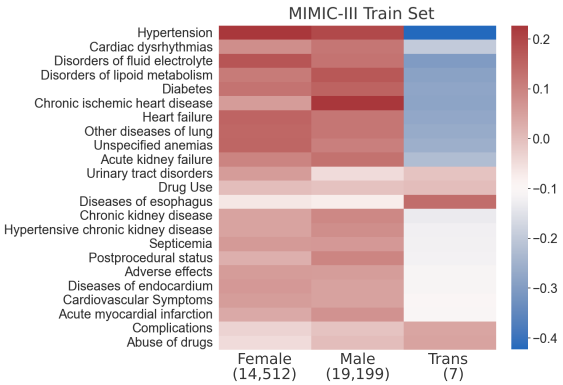


Figure 7.4. Original distribution of diagnoses per **gender** in MIMIC-III. Cell colors: Deviation from average probability. Numbers in parenthesis: Occurrences in the training set. Most diagnoses occur less often in transgender patients due to their very low sample count.

$$c_i = p_i - \frac{\sum_j^N p_j}{N} \tag{7.1}$$

where c_i is the value assigned to test group i, p is the (predicted) probability for a given diagnosis and N is the number of all test groups except i.

We choose this illustration based on the concept of partial dependence plots [Friedman, 2001] to highlight both positive and negative influence of a characteristic on model behavior. Since all test groups are based on the same patients and only differ regarding the characteristic at hand, even small differences in the averaged predictions can point towards general patterns that the model has learned.

7.5.1 Influence of Gender

Transgender mention leads to lower mortality and diagnoses predictions. Table 7.2 shows the mortality predictions of the three analyzed models with regard to the gender assigned in the text. While the predicted mortality risk for female and male patients lies within a small range, all models predict the mortality risk of patients that are described as transgender as lower than non-transgender patients. This is probably due to the relative young age of most transgender patients in the MIMIC-III training data, but can be harmful to older patients identifying as transgender at inference time.

Sensitivity to gender mention varies per model. Figure 7.3 shows the change in model prediction for each diagnosis with regard to the gender mention. The cells of the heatmap are the deviations from the average score of the other test cases. Thus, a red cell indicates that the model assigns a higher probability to a diagnosis for this gender group. We see that PubMedBERT is highly sensitive to the change of the patient gender, especially regarding transgender patients. Except from few diagnoses such as *Cardiac dysrhythmias* and *Drug Use / Abuse*, the model predicts a lower probability to diseases if the patient letter contains the transgender mention. The CORe and BioBERT models are less sensitive in this regard. The most salient deviation of the BioBERT model is a drop in probability of *Urinary tract disorders* for male patients, which is medically plausible due to anatomic differences [Tan and Chlebicki, 2016].

Patterns in MIMIC-III training data are partially inherited. In Figure 7.4 we show the original distribution of diagnoses per gender in the training data. Note that the deviations are about 10 times larger than the ones produced by the model predictions in Figure 7.3. This indicates that the models take gender as a decision factor, but only among others. Due to the very rare occurrence of transgender mentions (only seven cases in the training data), most diagnoses are underrepresented for this group. This is partially reflected by the model predictions, especially by PubMedBERT, as described above. Other salient patterns such as the prevalence of *Chronic ischemic heart disease* in male patients are only reproduced faintly.

	PubMedBERT	CORe	BioBERT
Female	**0.335**	0.239	0.119
Male	0.333	**0.245**	**0.121**
Transgender	0.326	0.229	0.117

Table 7.2. Influence of **gender** on mortality predictions. PubMedBERT assigns highest risk to female, the other models to male patients. Notably, all models decrease their mortality prediction for transgender patients.

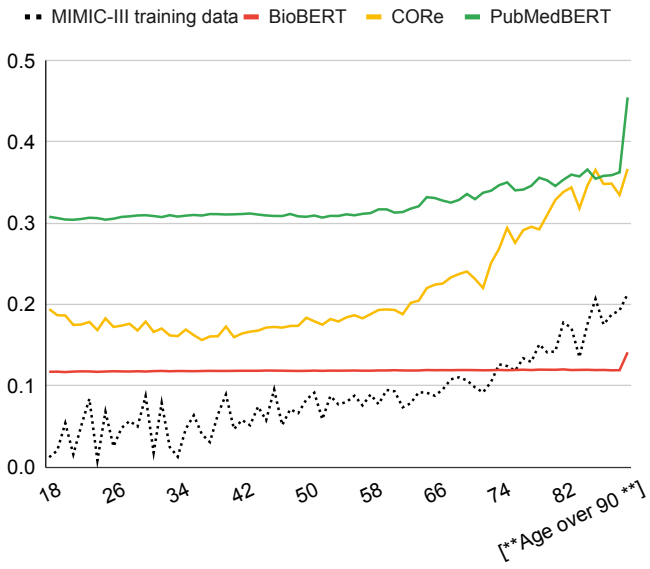


Figure 7.5. Influence of **age** on mortality predictions. X-axis: Simulated age; y-axis: predicted mortality risk. The three models are differently calibrated and only CORe is highly influenced by age.

7.5.2 Influence of Age

Mortality risk is differently influenced by age. Figure 7.5 shows the averaged predicted mortality per age for all models and the actual distribution from the training data (dotted line). We see that BioBERT does not take age into account when predicting mortality risk except for patients over 90. PubMedBERT assigns a higher mortality risk to all age groups with a small increase for patients over 60 and an even steeper increase for patients over 90. CORe follows the training data the most while also inheriting peaks and troughs in the data.

Models are equally affected by age when predicting diagnoses. We exemplify the impact of age on diagnosis prediction on eight outcome diagnoses in Figure 7.6. The dotted lines show the distribution of the diagnosis within an age group in the training data. The change of predictions regarding age are similar throughout the analyzed models with only small variations such as for *Cardiac dysrhythmias*. Some diagnoses are regarded more probable in older patients (e.g. *Acute Kidney Failure*) and others in younger patients (e.g. *Abuse of drugs*). The distributions per age group in the training data are more extreme, but follow the same tendencies as predicted by the models.

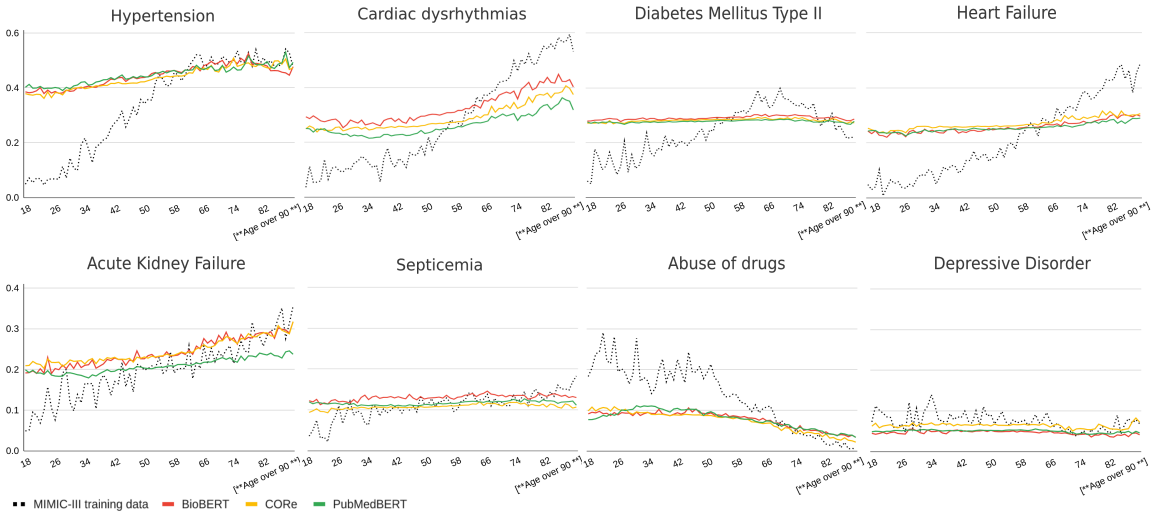


Figure 7.6. Influence of **age** on diagnosis predictions. The x-axis is the simulated age and the y-axis is the predicted probability of a diagnosis. All models follow similar patterns with some diagnosis risks increasing with age and some decreasing. The original training distributions (black dotted line) are mostly followed but attenuated.

Peaks indicate lack of number understanding. From earlier studies we know that BERT-based models have difficulties dealing with numbers in text [Wallace et al., 2019]. The peaks that we observe in some predictions support this finding. For instance, the models assign a higher risk of *Cardiac dysrhythmias* to patients aged 73 than to patients aged 74, because they do not capture that these are consecutive ages. Therefore, the influence of age on the predictions might solely be based on the individual age tokens observed in the training data.

7.5.3 Influence of Ethnicity

Mention of any ethnicity decreases prediction of mortality risk. Table 7.3 shows the mortality predictions when different ethnicities are mentioned and when there is no mention. We observe that the mention of any of the ethnicities leads to a decrease in mortality risk prediction in all models, with White and African American patients receiving the lowest probabilities.

Diagnoses predicted by PubMedBERT are highly sensitive to ethnicity mentions. Figure 7.7 depicts the influence of ethnicity mentions on the three models. Notably, the predictions of PubMedBERT are strongly influenced by ethnicity mentions. Multiple diagnoses such as *Chronic kidney disease* are more often predicted when there is no mention of ethnicity, while diagnoses like *Hypertension* and *Abuse of drugs* are regarded more likely in African American patients and *Unspecified anemias* in Hispanic patients. While the original training data in Figure 7.8 shows the same

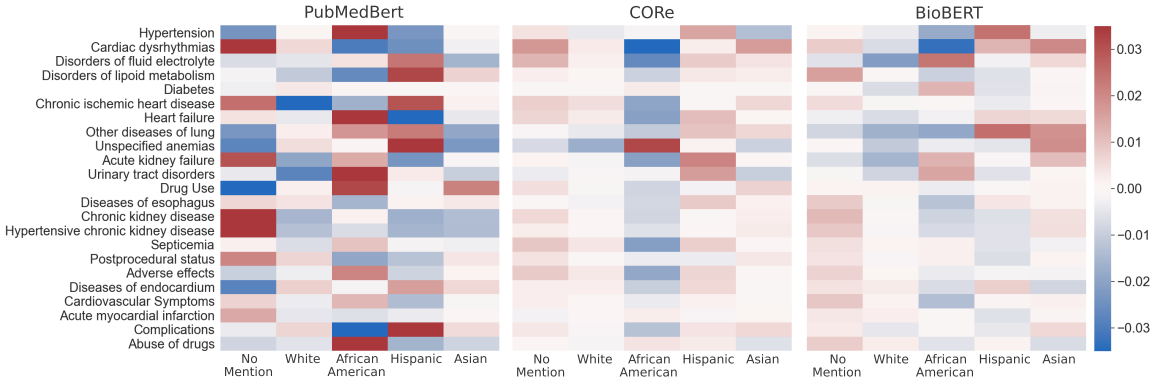


Figure 7.7. Influence of **ethnicity** on diagnosis predictions. Blue: Predicted probability for diagnosis is below-average; red: predicted probability above-average. PubMedBERT's predictions are highly influenced by ethnicity mentions, while CORe and BioBERT show smaller deviations, but also disparities on specific groups.

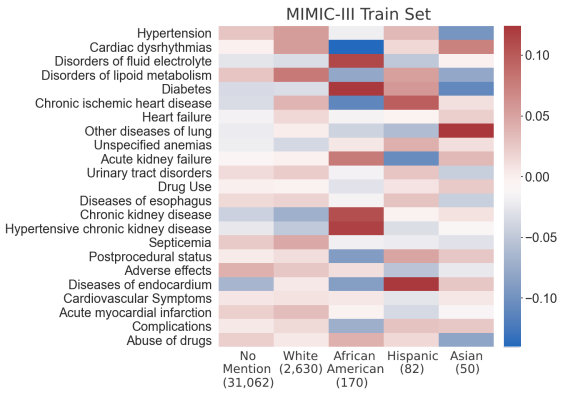


Figure 7.8. Original distribution of diagnoses per **ethnicity** in MIMIC-III. Cell colors: Deviation from average probability. Numbers in parenthesis: Occurrences in the training set. Both the distribution of samples and the occurrences of diagnoses are highly unbalanced in the training set.

strong variance among ethnicities, this is not inherited the same way in the CORe and BioBERT models. However, we can also observe deviations regarding ethnicity in these models.

African American patients are assigned lower risk of diagnoses by CORe and BioBERT. The heatmaps showing predictions of CORe and BioBERT reveal a potentially harmful pattern in which the mention of *African American* in a clinical note decreases the predictions for a large number of diagnoses. This pattern is found more prominently in the CORe model, but also in BioBERT. Putting these models into clinical application could result in fewer diagnostic tests to be ordered by physicians and, therefore, lead to disadvantages in the treatment of African American

patients. This is particularly critical as it would reinforce existing biases in health care [Nelson, 2002].

	PubMedBERT	CORe	BioBERT
No mention	**0.333**	**0.243**	**0.120**
White	0.329	0.235	0.119
African Amer.	0.329	0.239	0.116
Hispanic	0.331	0.237	0.118
Asian	0.330	0.238	0.118

Table 7.3. Influence of **ethnicity** on mortality predictions. The mention of an ethnicity decreases the predicted mortality risk. White and African American patients are assigned with the lowest mortality risk (gray-shaded).

7.6 Discussion

Model behaviors show large variance. The results described in 6.5.3 reveal large differences in the influence of patient characteristics throughout the models. The analysis shows that there is no overall *best* model, but each model has learned both useful patterns (e.g. age as a medical plausible risk factor) and potentially dangerous ones (e.g. decreases in diagnosis risks for minority groups). The large variance is surprising since the models have a shared architecture and are fine-tuned on the same data—they only differ in their pre-training. And while the reported AUROC scores for the models (Table 7.1) are close to each other, the variance in learned behavior show that we should consider in-depth analyses a crucial part of model evaluation in the clinical domain. This is especially important since harmful patterns in clinical NLP models are often fine-grained and difficult to detect.

Model scoring can obfuscate critical behavior. The analysis has shown that PubMedBERT which outperforms the other models in both mortality and diagnosis prediction by AUROC show larger sensitivity to mentions of gender and ethnicity in the text. Many of them, such as lower diagnosis risk assignment to African American patients, might lead to undertreatment. This is alerting since it particularly affects minority groups which are already disadvantaged by the health care system. It also shows that instead of measuring clinical models regarding rather abstract scores, looking at their potential impact to patients should be further emphasized. To communicate model behavior to medical professionals one possible direction could be to use behavioral analysis results as a part of clinical model cards as proposed by Mitchell et al. [2019].

Limitations of the proposed framework. Unlike other behavioral testing setups (see 7.2.2), results of our framework cannot be easily categorized into *correct* and *false* behavior. While increased risk allocations can be beneficial to a patient group due to doctors running additional tests, they can also lead to mistreatment or other diagnoses being overlooked. Same holds for the influence of rare mentions, such as *transgender*: One could argue that based on only seven occurrences in the training set the characteristic should have less impact on model decisions overall. However, some features e.g. regarding rare diseases should be recognized as important even if very infrequent. Since our models often lack such judgement, the decision about which patient characteristic to consider a risk factor and their impact on outcome predictions is still best made by medical professionals. Nevertheless, decision support systems can be beneficial if their behavior is transparently communicated. With this framework we want to take a step towards improving this communication.

7.7 Chapter Summary

We introduced a behavioral testing framework for the clinical domain to understand the effects of textual variations on model predictions. We applied this framework to three clinical LLMs to examine the impact of certain patient characteristics. Our evaluation demonstrates the concrete effects of these characteristics on the models' decisions. Our results show that the models—even with very similar AUROC scores—have learned very different behavioral patterns, some of them with high potential to disadvantage minority groups. With this study, we demonstrate the importance of model evaluation beyond common metrics especially in sensitive domains like health care. We recommend to use the proposed framework and the results of our evaluations for discussions with medical professionals. Being aware of specific model behavior and incorporating this knowledge into clinical decision making is a crucial step towards safe deployment of such models.

Furthermore, we take the results of this chapter as motivation towards more interpretable models that can be iteratively improved with medical professionals in the loop. This way, models could learn which patterns to stick to and which ones to discard. The next chapter introduces an approach for adapting LLMs towards these goals by embedding them into an interpretable architecture.

8 ProtoPatient: Interpretable Diagnosis Prediction Using Prototypical Networks and LLMs

Our studies in the previous chapters have shown that LLMs can achieve promising results in clinical tasks but also that they incorporate opaque patterns which are not always medically plausible. In clinical practice, however, such models must not only have a high accuracy, but provide doctors with interpretable and helpful results. In this chapter, we introduce ProtoPatient, a novel method based on prototypical networks and label-wise attention that presents a step towards both of these abilities. ProtoPatient makes predictions based on parts of the text that are similar to prototypical patients—providing justifications that doctors understand. We evaluate the model on two publicly available clinical datasets and show that it outperforms existing baselines. Quantitative and qualitative evaluations with medical doctors further demonstrate that the model provides valuable explanations for clinical decision support. With ProtoPatient we thus present a way to embed the strengths of LLMs into an architecture that satisfies the domain requirement for more transparent systems.

8.1 Introduction

Medical professionals are faced with a large amount of textual patient information every day. Clinical decision support systems (CDSS) aim to help clinicians in the process of decision-making based on such data. We specifically look at a subtask of CDSS, namely the prediction of clinical diagnosis from patient admission notes. When clinicians approach the task of diagnosis prediction, they usually take similar patients into account (from their own experience, clinic databases or by talking to their colleagues) who presented with typical or atypical signs of a disease. They then compare the patient at hand with these previous encounters and determine the patient's risk of having the same condition.

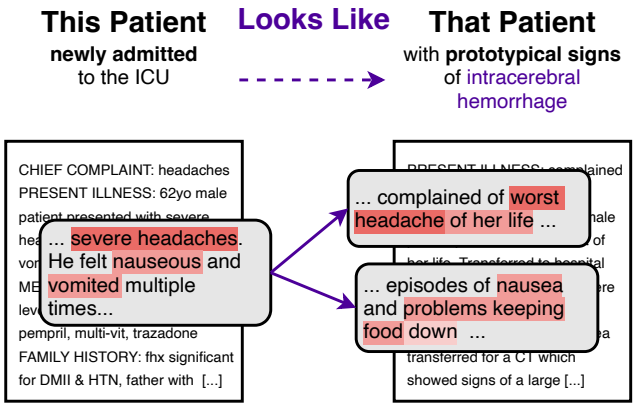


Figure 8.1. Basic concept of the ProtoPatient method. The model makes predictions for a patient (left side) based on the comparison to prototypical parts of earlier patients (right side).

In this chapter, we propose ProtoPatient, a deep neural approach that imitates this reasoning process of clinicians: Our model learns prototypical characteristics of diagnoses from previous patients and bases its prediction for a current patient on the similarity to these prototypes. This results in a model that is both inherently interpretable and provides clinicians with pointers to previous prototypical patients. Our approach is motivated by Chen et al. [2019] who introduced prototypical part networks (PPNs) for image classification. PPNs learn prototypical parts for image classes and base their classification on the similarity to these prototypical parts. We transfer this work into the text domain and apply it to the extreme multi-label classification task of diagnosis prediction. For this transfer, we apply an additional label-wise attention mechanism that further improves the interpretability of our method by highlighting the most relevant parts of a clinical note regarding a diagnosis.

While deep neural models have been widely applied to outcome prediction tasks in the past [Shamout et al., 2020], their black box nature remains a large obstacle for clinical application, as we have shown in Chapter 7. We argue that decision support is only possible when model predictions are accompanied by justifications that enable clinicians to follow a lead or to potentially discard predictions. With ProtoPatient we introduce an architecture that allows such decision support. Our evaluation on publicly available data shows that the model can further improve state-of-the-art performance on predicting clinical outcomes.

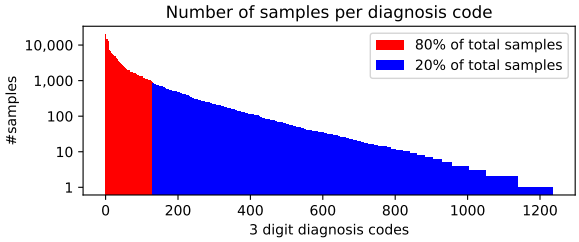


Figure 8.2. Distribution of ICD-9 diagnosis codes in the MIMIC-III training set.

8.2 Task: Diagnosis Prediction from Admission Notes

The task of outcome prediction from admission notes was introduced in Section 6.3 and assumes the following situation: A new patient p gets admitted to the hospital. Information about the patient is written into an admission note a_p. The goal of the decision support system is to identify risk factors in the text and to communicate these risks to the medical professional in charge. For outcome diagnosis prediction in particular, the underlying model determines these risks by predicting the likelihood of a set of diagnoses C being assigned to the patient at discharge.

Data. We evaluate our approach on the diagnosis prediction task from the clinical outcome prediction dataset introduced in Chapter 6. The data is based on the publicly available MIMIC-III database [Johnson et al., 2016]. It comprises de-identified data from patients in the Intensive Care Unit (ICU) of the Beth Israel Deaconess Medical Center in Massachusetts in the years 2001-2012. The data includes 48,745 admission notes written in English from 37,320 patients in total. They are split into train/val/test sets with no overlap in patients. The admission notes were created by extracting sections from MIMIC-III discharge summaries which contain information known at admission time such as *Chief Complaint* or *Family History*. The notes are labeled with diagnoses in the form of 3-digit ICD-9 codes that were assigned to the patients at discharge. On average, each patient has 11 assigned diagnoses per admission from a total set of 1266 diagnoses.

Challenges. Challenges surrounding diagnosis prediction can be divided into two main categories:

- **Predicting the correct diagnoses** The number of possible diagnoses is large (>1K) and, as shown in Figure 8.2, the distribution is extremely skewed. Since many diagnoses only have a few samples, learning plausible patterns is challenging. Further, each admission note describes multiple conditions, some being highly related while others are not. The text in admission notes is also highly context

dependent. Abbreviations like *SBP* (i.a. for *systolic blood pressure* or *spontaneous bacterial peritonitis*) have completely different meanings based on their context. Our models must capture these differences and enable users to check the validity of features used for a prediction.

- **Communicating risks to doctors** Apart from assigning scores to diagnoses, for a high stakes task such as diagnosis prediction, a system must be designed for medical professionals to understand and act upon its predictions. Therefore, models must provide faithful explanations for their predictions and give clues that enable further clinical reasoning steps by doctors. These requirements are challenging, since interpretability of models often come with a trade-off in their prediction performance [Arrieta et al., 2020].

8.3 Method

To address the challenges above, we propose a novel model architecture called ProtoPatient, which adapts the concept of prototypical networks [Chen et al., 2019] to the extreme multi-label scenario by using label-wise attention and dimensionality reduction. Figure 8.3 presents a schematic overview. We further show how our model can be efficiently initialized to improve both speed and performance.

8.3.1 Learning Prototypical Representations

We encode input documents a_p (p indexes patients) into vectors $\mathbf{v_p}$ with dimension D and measure their distance to a learned set of prototype vectors. Each prototype vector $\mathbf{u_c}$ represents a diagnosis $c \in C$ in the dataset. The prototype vectors are learned jointly with the document encoder so that patients with a diagnosis can best be distinguished from patients without it. As a distance measure we use the Euclidean distance $d_{pc} = ||\mathbf{v_p} - \mathbf{u_c}||_2$ which Snell et al. [2017] identified as best suited for prototypical networks. We then calculate the sigmoid σ of the negative distances to get a prediction $\hat{y}_{pc} = \sigma(-d_{pc})$, so that documents closer to a prototype vector get higher prediction scores. We define the loss L as the binary cross entropy (BCE) between $\hat{y}_{pc}$ and the ground truth $y_{pc} \in \{0, 1\}$.

$$L = \sum_p \sum_c BCE(\hat{y}_{pc}, y_{pc}) \tag{8.1}$$

Prototype initialization. Snell et al. [2017] define each prototype as the mean of the embedded support set documents. In contrast, we learn the label-wise prototype vectors end-to-end while optimizing the multi-label classification. This leads to better prototype representations since not all documents are equally representative of a

class, as taking the mean would suggest. However, using the mean of all support documents is a reasonable starting point. We set the initial prototype vectors of a class as $\mathbf{u_{c_{init}}} = \langle \mathbf{v_c} \rangle$, i.e. the mean of all document vectors $\mathbf{v_c}$ with class label c in the training set. We then fine-tune their representation during training. Initial experiments showed that this initialization leads to model convergence in half the number of steps compared to random initialization.

Contextualized document encoder. For the encoding of the documents, we choose a Transformer-based LLM due to their strengths demonstrated in Part I of this dissertation. For initializing the document encoder, we use the weights of a pre-trained language model. At the time of our experiments, the PubMedBERT [Gu et al., 2022] model reaches the best results on a range of biomedical NLP tasks. We thus initialize our document encoder with PubMedBERT weights[1] and further optimize it with a small learning rate during training.

8.3.2 Encoding Relevant Document Parts with Label-wise Attention

Since we face a multi-label problem, having only one joint representation per document tends to produce document vectors located in the center of multiple prototypes in vector space. This way, important features for single diagnoses can get blurred, especially if these diagnoses are rare. To prevent this, we follow the idea of prototypical part networks to select parts of the note that are of interest for a certain diagnosis. In contrast to Chen et al. [2019], we use an attention-based approach instead of convolutional filters, since attention is an effective way for selecting relevant parts of text. For each diagnosis c, we learn an attention vector $\mathbf{w_c}$. To encode a patient note with regard to c, we apply a dot product between $\mathbf{w_c}$ and each embedded token $\mathbf{g_{pj}}$, where j is the token index. We then apply a softmax.

$$s_{pcj} = softmax(\mathbf{g_{pj}^T}\, \mathbf{w_c}) \tag{8.2}$$

We use the resulting scores s_{pcj} to create a document representation $\mathbf{v_{pc}}$ as a weighted sum of token vectors.

$$\mathbf{v_{pc}} = \sum_j s_{pcj}\, \mathbf{g_{pj}} \tag{8.3}$$

This way, the document representation for a certain diagnosis is based on the parts that are most relevant to that diagnosis. We then measure the distance $d_{pc} = ||\mathbf{v_{pc}} - \mathbf{u_c}||_2$ to the prototype vector $\mathbf{u_c}$ based on the diagnosis-specific document representation $\mathbf{v_{pc}}$.

[1] Model weights from: https://huggingface.co/microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext

Attention initialization. The label-wise attention vectors $\mathbf{w_c}$ determine which tokens the final document representation is based on. Therefore, when initializing them randomly, we start our training with document representations which might carry little information about the patient and the corresponding diagnosis. To prevent this cold start, we initialize the attention vectors $\mathbf{w_{c_{init}}}$ with tokens informative to the diagnosis c. This way, at training start, these tokens reach higher initial scores s_{pcj}. We consider tokens $\tilde{t}$ informative that surpass a TF-IDF threshold of h. We then use the average of all embeddings $\mathbf{g_{c\tilde{t}}}$ from $\tilde{t}$ in documents corresponding to the diagnosis.

$$\mathbf{w_{c_{init}}} = \langle \mathbf{g_{c\tilde{t}}} \rangle \tag{8.4}$$

with $\tilde{t} = t : \text{tf-idf}(t) > h$. We found h=0.05 suitable to get 5-10 informative tokens per diagnosis.

8.3.3 Compressing Representations

Label-wise attention vectors for a label space with more than a thousand labels lead to a considerable increase in model parameters and memory load. We compensate this by reducing the dimensionality D of vector representations used in our model. We add a linear layer after the document encoder that both reduces the size of the document embeddings and acts as a regularizer, compressing the information encoded for each document. We find that reducing the dimensionality by one third ($D = 256$) leads to improved results compared to the full-size model, indicating that more dense representations are beneficial to our setup.

8.3.4 Presenting Prototypical Patients

For retrieving prototypical patients $\mathbf{v'_c}$ for decision justifications at inference time, we simply take the label-wise attended documents from the training data that are closest to the diagnosis prototype. By presenting their distances to the prototype vector, we can provide further insights about the general variance of diagnosis presentations. Correspondingly, we can also present patients with atypical presentation of a diagnosis by selecting the ones furthest away from the learned prototype.

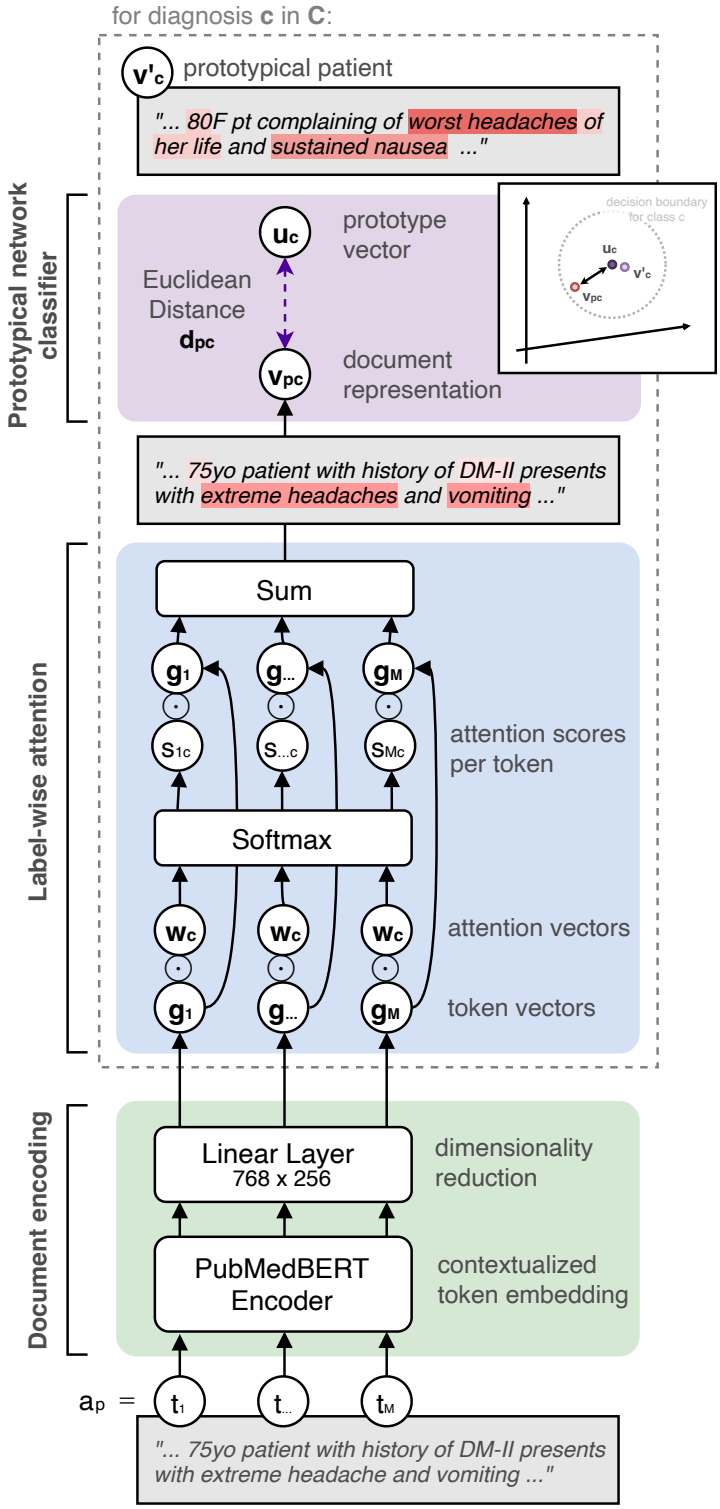


Figure 8.3. Schematic view of the ProtoPatient method. Starting at the bottom, document tokens get a contextualized encoding and are then transformed into a label-wise document representation $\mathbf{v_{pc}}$. The classifier simply considers the distance of this representation to a learned prototypical vector $\mathbf{u_c}$. The prototypical patient $\mathbf{v'_c}$ is the training example closest to the prototypical vector.

8.4 Evaluating Diagnosis Predictions

8.4.1 Experimental Setup

Baselines. We compare ProtoPatient to hierarchical attention models and to LLMs pre-trained on (bio)medical text, representing two state-of-the-arts approaches for ICD coding and outcome prediction tasks, respectively.

- **Hierarchical attention models** Hierarchical Attention Networks (**HAN**) were introduced by Yang et al. [2016]. They are based on bidirectional Gated Recurrent Units with attention applied on both the sentence and token level. Baumel et al. [2018] built **HA-GRU** upon this concept using only sentence-wise attention, while adding a label-wise attention scheme comparable to ProtoPatient. Dong et al. [2021] further show that pre-initialized **label embeddings** learned from ICD code co-occurrence improves results for both approaches. We thus evaluate the models with and without label embeddings.[2]

- **Transformers pre-trained on in-domain text** Alsentzer et al. [2019] applied clinical language model fine-tuning on two LLMs based on the BioBERT model [Lee et al., 2020]. **ClinicalBERT** was trained on all clinical notes in the MIMIC-III database, and **DischargeBERT** on all discharge summaries. They belong to the most widely used clinical language models and achieve high scores on multiple clinical NLP tasks. The **CORe** model introduced in Chapter 6 is also based on BioBERT, but further pre-trained with an objective specific to patient outcomes, which achieved higher scores on clinical outcome prediction tasks. Tinn et al. [2021] introduced **PubMedBERT** which was, in contrast to the other models, trained from scratch on articles from PubMed Central with a dedicated vocabulary. At the time of experimentation, it was the best performing approach on the BLURB [Gu et al., 2022] benchmark.

Training. We train all baselines on the dataset introduced in Section 8.2. For training HAN and HA-GRU, we use the code and best performing hyperparameters as provided by Dong et al. [2021]. We further apply label embeddings to the HAN and HA-GRU network as proposed by Dong et al. [2021]. In particular, we use the pre-initialized embeddings provided by the authors. Since they use a larger label set, we map their embedding vectors to the ICD-9 groups we use in our study. The mapping is done by averaging all subcodes for one group. If no code is available for an ICD-9 group, we use a randomly initialized vector. For training the Transformer-based

[2]Note that Dong et al. [2021] also propose the H-LAN model, which is a combination of HAN and HA-GRU using label-wise attention on sentence and token level. However, the model is only applicable to smaller label spaces (<100) due to its memory footprint and thus cannot be evaluated on our task.

	AUROC macro	AUROC micro	AUPRC macro
HAN [Yang et al., 2016]	83.38 ±0.13	96.88 ±0.04	13.56 ±0.01
HAN + Label Emb [Dong et al., 2021]	83.49 ±0.18	96.87 ±0.12	13.07 ±0.14
HA-GRU [Baumel et al., 2018]	79.94 ±0.57	96.65 ±0.12	9.52 ±1.01
HA-GRU + Label Emb [Dong et al., 2021]	80.54 ±1.67	96.67 ±0.22	10.33 ±1.70
ClinicalBERT [Alsentzer et al., 2019]	80.95 ±0.16	94.54 ±0.93	11.62 ±0.64
DischargeBERT [Alsentzer et al., 2019]	81.17 ±0.30	94.70 ±0.48	11.24 ±0.88
CORe [van Aken et al., 2021a]	81.92 ±0.09	94.00 ±1.10	11.65 ±0.78
PubMedBERT [Tinn et al., 2021]	83.48 ±0.21	95.47 ±0.22	13.42 ±0.57
Prototypical Network	81.89 ±0.22	95.23 ±0.01	9.94 ±0.36
ProtoPatient	86.93 ±0.24	**97.32** ±0.00	**21.16** ±0.21
ProtoPatient + Attention Init	**87.93** ±0.07	97.24 ±0.02	17.92 ±0.65

Table 8.1. Results in % AUROC for diagnosis prediction task (1266 labels) based on MIMIC-III data. The ProtoPatient model outperforms the baselines in micro AUROC and AUPRC. The attention initialization further improves the macro AUROC. Label Emb: Label Embeddings. Attention Init: Attention vectors initialized as described in Section 8.3.2. ± values are standard deviations.

LLMs and ProtoPatient, we use hyperparameters that perform best for BERT-based models in earlier experiments and additionally optimize the learning rate and number of warm up steps with a grid search. We further truncate the notes to a context size of 512.

Since we work with 1266 labels, the label-wise attention calculations limit the batch size that fits into memory. Therefore, we use a batch size of 20 for all models without label-wise attention, 10 for label-wise attention models reduced to a dimensionality of 256, and 5 for the others. Initial experiments showed that the batch sizes have no influence on model performance in our experiments, only on memory consumption and training duration.

We choose different learning rates for the document encoder weights and the prototype and label-wise attention vectors. Since we expect the encoder weights from the pre-trained LLMs to be already well aligned with clinical language, we choose a small learning rate between 5e-04 and 5e-06. The prototypical diagnosis vectors and the label-wise attention vectors need more adjustments to enable the classification task, so we search in a range of 5e-02 and 5e-04. We further apply an AdamW [Loshchilov and Hutter, 2017] optimizer and a linear learning rate scheduler with a warm-up period of 1K to 5K steps. We provide the best hyperparameters per model in the public code repository. We further report the scores of all models as an average over three runs with different seeds.

Ablation studies. ProtoPatient combines three strategies: Prototypical networks, label-wise attention and dimensionality reduction. We conduct ablation studies to measure the impact of each strategy. To this end, we apply both label-wise attention

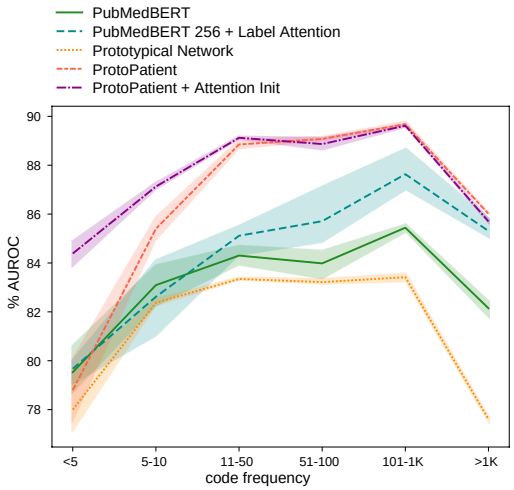


Figure 8.4. Macro AUROC scores regarding the frequency of ICD-9 codes in the training set. ProtoPatient models show the largest performance gain in rare codes ($\leq$100 samples). Attention initialization leads to large improvement for extremely rare ones ($<$10 samples).

and dimensionality reduction to a PubMedBERT model using a standard classification head. We further train a prototypical network without label-wise attention and ProtoPatient with different dimension sizes. The results are found in Table 8.2.

Transfer to second data set. Clinical text data varies from clinic to clinic. We want to test whether the patterns learned by the models are transferable to other data sources than MIMIC-III. We use another publicly available dataset from the i2b2 De-identification and Heart Disease Risk Factors Challenge [Stubbs and Uzuner, 2015] further processed into admission notes as described in Section 6.5.4. The data consists of 1,118 admission notes labeled with the ICD-9 codes for *chronic ischemic heart disease*, *obesity*, *hypertension*, *hypercholesterolemia* and *diabetes*. We evaluate models without fine-tuning on the new data to simulate a model transfer to another clinic. The resulting scores are reported in Table 8.3.

8.4.2 Results

We present the results of all models on the diagnosis prediction task in Table 8.1. In addition, we show the macro AUROC score across codes depending on their frequency in the training set in Figure 8.4. We summarize the main findings as follows.

ProtoPatient outperforms previous approaches. The results show that ProtoPatient achieves the best scores among all evaluated models. Pre-initializing the attention vectors further improves the macro AUROC score. Ablation studies show

	AUROC macro	AUROC micro	AUPRC macro
Dimensionality reduction			
ProtoPatient 768	83.56 ±0.17	96.65 ±0.03	14.36 ±0.16
ProtoPatient 256	**86.93** ±0.24	**97.32** ±0.00	**21.16** ±0.21
Prototypical LLM vs. LLM			
ProtoPatient 768	83.56 ±0.17	**96.65** ±0.03	14.36 ±0.16
PubMedBERT 768 + Label Attention	**84.10** ±0.25	**96.66** ±0.17	**19.74** ±1.27
Label-wise attention			
PubMedBERT 256	83.61 ±0.04	95.76 ±0.05	13.35 ±0.25
PubMedBERT 256 + Label Attention	84.68 ±0.52	96.86 ±0.14	17.15 ±1.52
ProtoPatient 256	**86.93** ±0.24	**97.32** ±0.00	**21.16** ±0.21

Table 8.2. Ablation studies comparing different dimension sizes and how a standard LLM (PubMedBERT) performs with additional label-wise attention. Smaller dimension sizes benefit ProtoPatient, while the effect is less notable on PubMedBERT. Adding label-wise attention, however, increases PubMedBERT results clearly. Overall, the combination of prototypical network, label-wise attention, and reduced dimension in ProtoPatient reaches the best results.

that all components play a role in improving the results. A prototypical network without label-wise attention is not able to capture the extreme multi-label data. PubMedBERT using a standard classification head also benefits from label-wise attention, but not to the same extent. Combining prototypical networks and label-wise attention thus brings additional benefits. The choice of dimension size is another important factor. Using 768 dimensions (the standard BERT base size) appears to lead to over-parameterization in the attention and prototype vectors. Using 256 dimensions also improves generalization which is shown in producing the best results on the i2b2 data set in Table 8.3.

Improvements for rare diagnoses. Figure 8.4 shows that the AUROC improvements are particularly large for codes that are rare ($\leq$50 times) in the training set. Prototypical networks are known for their few-shot capabilities [Snell et al., 2017] which also prove useful in our scenario with mixed label frequencies. For extremely rare codes that appear less than ten times, the attention initialization described in Section 8.3.2 further improves results. This indicates that the randomly initialized attention vectors need at least a number of samples to learn the most important tokens, and that pre-initializing them can accelerate this process.

PubMedBERT and HAN are the best baselines. The pre-trained PubMedBERT and the HAN model achieve the highest scores among the baselines. Interestingly, PubMedBERT outperforms the LLMs pre-trained on clinical text. This indicates that training from scratch with a domain-specific vocabulary is beneficial

	AUROC macro	AUROC micro	AUPRC macro
PubMedBERT	82.11 ±0.12	85.48 ±0.64	84.38 ±0.54
PubMedBERT 256 + Label Attention	79.78 ±5.30	83.43 ±4.54	84.70 ±2.84
Prototypical Network	69.65 ±0.22	74.31 ±0.19	78.53 ±0.19
ProtoPatient 768	85.28 ±0.49	88.63 ±0.43	87.78 ±0.10
ProtoPatient	**87.38** ±0.20	**90.63** ±0.23	**89.72** ±0.24
ProtoPatient + Attention Init	86.72 ±1.52	89.84 ±1.16	**89.71** ±1.20

Table 8.3. Performance on a second data set based on clinical notes from the i2b2 challenge [Stubbs and Uzuner, 2015]. Note that the baseline AUPRC is much higher for this task than for the task based on MIMIC-III. ProtoPatient models reach the highest scores, indicating that they are more robust towards changes in text style than the PubMedBERT baselines. The PubMedBERT model with label-wise attention, in particular, shows quite inconsistent results regarding different seeds.

for the task. The scores of the HAN model further emphasize the importance of label-wise attention. The addition of label embeddings to HAN and HA-GRU, however, does not add significant improvements in our case.

8.5 Evaluating Interpretability

We evaluate the interpretability of ProtoPatient with quantitative and qualitative analyses as follows.

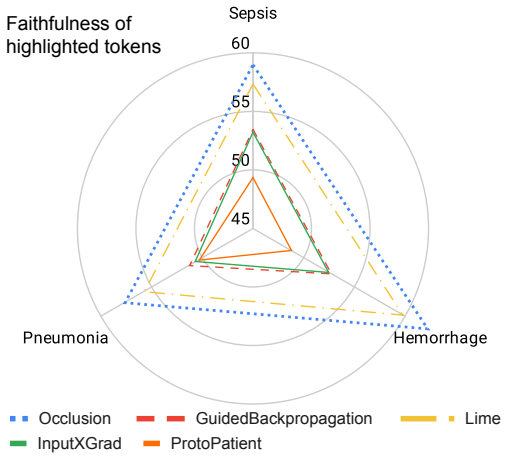


Figure 8.5. Evaluating faithfulness of highlighted tokens. Lower scores indicate more faithful explanations. ProtoPatient's token highlights are part of the model decision and thus more faithful than post-hoc explanations.

Diagnosis	15 most attended words - with medical relation to diagnosis
Sepsis	1. **hypotension** symptom, 2. **sepsis** descriptor, 3. **fever** symptom, 4. **hypotensive** symptom, 5. **fevers** symptom, 6. **septic** descriptor, 7. **lactate** indicator, 8. **shock** descriptor, 9. **bacteremia** descriptor, 10. **febrile** symptom, 11. **vancomycin** medication, 12. **SBP** risk factor, 13. **levophed** medication, 14. **swelling** symptom, 15. **cirrhosis** risk factor
Intracerebral Hemorrhage	1. **hemorrhage** descriptor, 2. **bleed** descriptor, 3. **headache** symptom, 4. **ICH** descriptor, 5. **IPH** descriptor, 6. **CT** diagnostic, 7. **weakness** symptom, 8. **stroke** descriptor, 9. **brain** descriptor, 10. **intracranial** descriptor, 11. **hemorrhagic** descriptor, 12. **intraventricular** descriptor, 13. **hemorrhages** descriptor, 14. **hemiparesis** symptom, 15. **aphasia** symptom
Pneumonia	1. **pneumonia** descriptor, 2. **cough** symptom, 3. **PNA** descriptor, 4. **COPD** risk factor, 5. **infiltrate** symptom, 6. **distress** complication, 7. **fever** symptom, 8. **breath** *ambiguous*, 9. **hypoxia** symptom, 10. **sputum** symptom, 11. **respiratory** complication, 12. **sepsis** complication, 13. **SOB** symptom, 14. **consolidation** symptom, 15. **CAP** descriptor

Table 8.4. Words from the test set with the highest attention scores assigned by ProtoPatient. All words are directly related to the diagnoses and mostly describe symptoms or direct descriptors (in various forms). The highlights can, therefore, help doctors to quickly identify important parts within a note and to compare them to prototypical parts.

Quantitative study on faithfulness. Faithfulness describes how explanations correspond to the inner workings of a model, a property essential to their usefulness. We apply the explainability benchmark introduced by Atanasova et al. [2020] to compare the faithfulness of ProtoPatient's token highlights to post-hoc explanation methods. Following the benchmark, faithfulness is measured by incrementally masking highlighted tokens, expecting a steep drop in model performance if the tokens are indeed relevant to the model prediction.

The framework evaluates different methods that output saliencies indicating token importance for a model decision. The evaluation masks the most salient tokens via multiple thresholds and measures the model's performance for each one. Thresholds are going from masking only the top 10% of salient tokens in steps of 10pp until 100% of tokens are masked. The final faithfulness score is then calculated as the area under the curve of model performance over all thresholds. As a performance measure, we choose macro AUROC to stay consistent with the rest of our experiments. We compare tokens highlighted by ProtoPatient's label-wise attention vectors to four common post-hoc explanation methods, namely Lime [Ribeiro et al., 2016], Occlusion [Zeiler and Fergus, 2014], InputXGradient [Kindermans et al., 2016], and Gradient Backpropagation [Springenberg et al., 2015]. We apply these methods to the PubMedBERT baseline, corresponding to a typical post-hoc explanation approach for an otherwise black box model.

Due to the high computational costs of the evaluation, we limit our analyses to three diagnoses with a high severity to the ICU: *Sepsis*, *intracerebral hemorrhage* and *pneumonia*. Figure 8.5 shows the results, for which lower scores mean more faithful explanations (i.e. a steeper drop in model performance). We see that ProtoPatient's

explanations reach the lowest scores for all three labels, proving that they are more faithful than the post-hoc explanations. This is a result of the interpretable structure of ProtoPatient, in which model decisions are directly based on the highlighted parts.

Finding most relevant words per diagnosis. We want to examine which parts of the clinical notes are highlighted by ProtoPatient per diagnosis. To that end, we collect the tokens with the highest attention scores over all training samples per label. We again use the three diagnoses *sepsis*, *intracerebral hemorrhage* and *pneumonia* for a closer analysis. We further map the tokens to their corresponding words. We then let doctors define the words' medical relations to understand which features the model considers important. Table 8.4 shows that the most attended words are mainly symptoms or descriptors of the condition at hand, which meets the objective of ProtoPatient to point doctors to relevant parts of a note.

Manual analysis by medical doctors. We conduct a manual analysis with two medical doctors (one specialized, one resident) to understand whether highlighted tokens and prototypical patients are helpful for their decisions. They used a demo application of ProtoPatient and analyzed 20 random patient letters with 203 diagnoses in total. The results are shown in Table 8.5. The doctors first identified the principal diagnoses and then rated the corresponding prototypical patients presented by the model. Note that some patients have more than one principal diagnosis. In 21 of 23 cases, the prototypical samples were showing typical signs of the respective diagnosis and 17 of them were rated as helpful for making a diagnosis decision. Cases in which they were not helpful included very rare conditions or ones with a strong difference to the specific case. They further analyzed the highlighted tokens for all diagnoses and found that they contained mostly relevant information in 150 cases. Examples of highlighted risk factors judged as plausible were *obesity* known to relate to *diabetes type II*, *untreated hypertension* to *heart failure* or a medication history of *anticoagulant coumadin* to *atrial fibrillation*. They also identified cases in which the highlighted tokens were partially or hardly relevant. In these cases, the highlighted tokens often included stop words or punctuation, indicating that the attention vector failed to learn relevant tokens. This was mainly observed in very frequent diagnoses such as *hypertension* or *anemia*, which corresponds to the lower model performance on these conditions (see Figure 8.4). This is because conditions very common in the ICU are often either not indicated in the text of the clinical note or not labeled, as described in 6.6.2, so that the model cannot learn clear patterns regarding their relevant tokens.

Analysis of prototypical patient cases (principal diagnoses)			
Q1: Prototypical patient shows typical clinical signs			
yes		no	
21		2	
Q2: Highlighted prototypical parts are relevant			
mostly	partially	hardly	
21	2	0	
Q3: Prototypical patient is helpful for diagnosis decision			
yes		no	
17		6	
Analysis of highlighted parts (all diagnoses)			
Q4: Highlighted tokens are relevant for diagnosis (i.e. describe diagnosis, symptoms or risk factors)			
	mostly	partially	hardly
TPs	78	3	7
FPs	50	12	9
FNs	22	10	12
Q5: Important tokens are missing from highlights			
	yes	no	
TPs	17	71	
FPs	13	58	
FNs	2	42	

Table 8.5. Results of the manual analysis conducted by medical doctors on ProtoPatient outputs. The prototypical patients were analyzed for the principal diagnoses only, while the highlighted parts of the patient letter at hand were analyzed for all diagnoses. Q1..5 denote the questions answered regarding each patient case.

Admission note	Relevant parts of admission note		Parts of prototypical patient notes
PRESENT ILLNESS: Patient is a 35-year-old male pedestrian struck by a bicycle from behind with positive loss of consciousness for 6 minutes at the scene after landing on his head. At arrival at ER patient was confused, had multiple contusions noted on a head CT scan including bilateral frontal and right temporal contusions. His cervical spine and abdominal examinations were negative radiologically. The patient was then transferred to the Emergency Room. Patient had several episodes of vomiting during flight and during the trauma workup. He was assessed and was intubated for airway protection. The patient was given coma score of 9 upon initial assessment. Patient remaining hemodynamically stable throughout the transfer and throughout the workup in the ED. [. . .]	struck by a bicycle . . . loss of consciousness for 6 minutes . . . coma score 9 . . .	⟶	**cerebral hemorrhage** loss of consciousness . . . struck by vehicle . . . with a gcs of 10 . . .
	head CT scan . . . bilateral contusions . . . hemodynamically stable . . .	⟶	**skull fracture** head wound . . . right and left contusions . . . stable blood circulation . . .
	transferred to Emergency Room . . . several episodes of vomiting . . .	⟶	**shock** patient had multiple episodes of vomiting during the day . . .
	patient was confused . . . intubated for airway protection . . .	⟶	**acute respiratory failure** patient was disoriented . . . later intubated for protection. . .

Table 8.6. Exemplary output of ProtoPatient. The model identifies parts in an admission note that are similar to (i.e. *"look like"*) parts from prototypical patient notes seen during training leading to the prediction of this diagnosis.

8.6 Related Work

Diagnosis prediction from clinical notes. Predicting diagnosis risks from clinical text has been studied using different methods. Fakhraie [2011] analyzed the predictive value of clinical notes with bag of words and word embeddings. Jain et al. [2019] experimented with adding attention modules to recurrent neural models. Recently, the use of Transformer-based LLMs for diagnosis prediction has outperformed earlier approaches. We applied BERT-based models further pre-trained on clinical cases to predict patient outcomes in Chapter 6. However, the black box nature of these models hinders their application in clinical practice. We thus introduce ProtoPatient, which uses representations from LLMs but provides interpretable predictions.

Prototypical networks for few-shot learning. Prototypical networks were first introduced by Snell et al. [2017] for few-shot learning. They initialized prototypes as centroids of support samples per episode and applied the approach to image classification. Sun et al. [2019a] adapted the approach to text documents with hierarchical attention layers. Related approaches based on prototypical networks have been used for multiple few-shot text classification tasks [Wen et al., 2021, Zhang et al., 2021b, Ren et al., 2020, Deng et al., 2020, Feng et al., 2023]. In contrast, we do not train our model in a few-shot scenario using episodic learning. However, our model shows related capabilities by improving results for diagnoses with few available samples.

Prototypical networks for interpretable models. Chen et al. [2019] used prototypical networks in a different setup to build an interpretable model for image classification. To this end, they learn prototypical parts of images to mimic human reasoning. We adapt their idea and show how to apply it to clinical natural language. Comparably, Ming et al. [2019] and Das et al. [2022] applied the concept of prototypical networks to text classification and showed how prototypical texts help to interpret predictions. In contrast to their work and following Chen et al. [2019], we identify prototypical *parts* rather than whole documents by using label-wise attention. This makes interpreting results easier and enables multi-label classification with over a thousand labels.

Label-wise attention. Mullenbach et al. [2018] introduced label-wise attention for clinical text with the CAML model. Since then, the method has been further improved by hierarchical attention approaches [Baumel et al., 2018, Yang et al., 2016, Dong et al., 2021]. Label-wise attention has mainly been used for ICD coding, a task related to diagnosis prediction that differs in the input data: ICD coding is done on notes that describe the whole stay at a clinic. In contrast, outcome diagnosis prediction uses admission notes as input and identifies diagnosis *risks* rather than the diagnoses already mentioned in the text. Our method—combining prototypical networks with label-wise attention—is particularly focused on detecting and highlighting those risks to enable clinical decision support.

8.7 Discussion

8.7.1 Reflection on the Challenges

Rudin [2019] urges to stop explaining black boxes and to build interpretable models instead. With ProtoPatient we introduce a model with a simple decision process—*this patient looks like that patient*—that is understandable to medical professionals and inherently interpretable. An exemplary output is shown in Table 8.6. Our results indicate that the model is able to deal with contextual text in clinical notes, e.g. when identifying *SBP* as a risk factor for sepsis in 8.5. In addition, it improves results on rare diagnoses, which are especially challenging for doctors to detect due to the lack of experience and sensitivity towards their signs. Overall, our approach demonstrates that interpretability can be improved without compromising performance. The modularity of the prototype vectors further allows clinicians to modify the model even after training. This can be done by adding prototypes whenever a new condition is found, or by directly defining certain patients as prototypical for the system.

8.7.2 Limitations of this Work

Our model currently learns relations between diagnoses only indirectly, due to the label-wise nature of the classification. However, considering relations or conflicts between diagnoses is an important part of clinical decision-making. One way to include such relations is the addition of a loss term incorporating diagnosis relations, as proposed by Mullenbach et al. [2018]. Another limitation is that the current model only considers one prototype per diagnosis, even though most diagnoses have multiple presentations, varying among patient groups. We, therefore, propose further research towards including multiple prototypes into the system.

8.8 Chapter Summary

In this chapter, we presented ProtoPatient which enables interpretable outcome diagnosis prediction from text. The proposed approach enhances existing methods in their prediction capability—especially for rare classes—and presents benefits to doctors by highlighting relevant parts in the text and pointing towards prototypical patients. The modularity of prototypical networks is a promising characteristic that should be further explored in future research. Prototypes could be added manually by medical professionals based on patients they consider prototypical. Another approach would be to initialize prototypes from medical literature and compare them to those learned from patients.

Overall, we showed that large language models can be used as building blocks in systems that incorporate characteristics beneficial to domain-specific needs. Prototypical networks have been demonstrated to be one of such systems in which LLMs can be embedded to. Our findings indicate that we can use such wrapping systems for adapting LLMs to even more specific domain requirements while benefiting from their ability to effectively encode language.

This chapter concludes the second part of the dissertation. In this part, we presented approaches to adapt LLMs to the clinical domain. We first showed that in-domain pre-training data with similar language style is an important factor for the performance of LLMs in the clinical domain and that task-specific data often lacks the variety to cover long tail expressions. This motivated research toward incorporating further resources into our models. To this end, we presented two approaches using unlabeled text resources as input for pre-training and additional knowledge from the medical coding system ICD for fine-tuning. We further studied adaptation with regard to the transparency needs of domain experts and introduced a clinical behavioral testing framework to improve the communication of model capabilities. The framework also highlights the need for interpretable systems in high stakes settings such as diagnosis prediction. Following this, we finally showed that such interpretable systems can be applied in combination with LLMs using architectures such as ProtoPatient.

9 Conclusion and Future Work

9.1 Summary of Contributions

In this dissertation, we analyzed how large language models can be applied and adapted to address the requirements and challenges posed by specialized domains. As a prerequisite, we explored capabilities and inner workings of LLMs. We showed that missing paradigmatic context and the lack of world knowledge hinders earlier approaches in capturing the granularities of specialized domains. We then analyzed the layer transformations of LLMs and demonstrated how such models can be examined and interpreted for a better understanding of their abilities. In the second part of this dissertation, we presented different methods for adapting LLMs to specialized domains and tasks. As an exemplary domain, we investigated the adaptation of LLMs to clinical text documents, in particular to patient notes. This research included the definition and framing of domain-specific tasks and datasets. The adaptations we presented are using LLMs as building blocks to fulfill identified domain requirements, such as the incorporation of domain-knowledge from un- or semi-labeled data and the need for explainable and justifiable solutions. In the following, we summarize the contributions presented in this work.

Analyzing inner workings and behavior of large language models. As the first step to understand the potential of LLMs, we presented an ensemble-based method for a joint error analysis of machine learning systems. This way, we were able to identify that a large source of errors was grounded in the lack of world knowledge encoded in earlier models. We also found many errors based on missing contextualization. Large language models based on Transformers encode text with regard to its context and are commonly pre-trained on a large amount of unlabeled training data, which allows to incorporate world knowledge that is missing from the sparse domain-specific labeled data. In an in-depth analysis of LLM layers, we were able to identify the phases of transformations happening within Transformer-based language

models. Our findings supported the hypothesis that Transformer-based LLMs encode knowledge about language and relations between entities learned during pre-training in their lower layers, which allows the later layers to use this knowledge for solving domain-specific tasks. Due to these characteristics LLMs show such large potential for the use in specialized domains.

Improving the general understanding of large language models is important for further development of their technology. However, for their save application in specialized domains, we require additional domain-specific analyses of their behavior targeted to domain experts. Therefore, in the second part of this dissertation, we presented a framework for behavioral testing of NLP models adapted to the clinical domain and to be used for communicating model behavior and risks to medical professionals. The framework can be extended to further use cases and requirements of the particular end users.

Identification and creation of tasks and datasets. The second part of this dissertation addresses the adaptation of LLMs to the clinical domain. In order to do understand how NLP systems can benefit clinical practice, we first identified common tasks in collaboration with medical professionals. The first task we analyzed is the classification of assertions in clinical text. This information extraction task requires the model to identify terms that confirm, negate or conjecture a medical condition. Due to the missing variety in existing data, we annotated 5,000 assertions from different types of clinical patient notes that we published to encourage further research.

Building up on this information extraction scenario, we further analyzed how LLMs can be applied for clinical decision support. Together with medical professionals, we developed a *clinical outcome prediction* task with an *admission to discharge* objective. We used publicly available data and transformed it according to domain-specific requirements spanning four common outcome prediction tasks that can be approached by deep learning models such as LLMs. This benchmark is also made public and can be used to test future model derivatives.

Adaptations to domain-specific use cases. We presented different ways of adapting LLM-based models to the clinical domain. First, we showed the influence of language model pre-training based on different data sources in Chapter 5. Models pre-trained on in-domain data outperformed general domain data by large margins; the closer the pre-training text to the final documents, the better the downstream performance. On this basis, we introduced a novel *clinical outcome pre-training*, a use case specific pre-training objective that harnesses semi-structured unlabeled clinical documents to integrate knowledge about patient trajectories into the model. This additional pre-training data can come both from patient data (clinical experience) and from literature (verified medical knowledge). We further proposed a method to incorporate hierarchical information from domain-specific ontologies into model

weights by using ICD hierarchies for multi-label classification. Both these methods introduced in Chapter 6 represent approaches to complement the models with additional domain knowledge that is usually not fully available from the (oftentimes scarce) labeled fine-tuning data.

In Chapter 8, we then showed how to use LLMs as building blocks in systems that are even more customized to the requirements of the use case at hand. We investigated the use case of clinical diagnosis prediction, which requires not only accurate predictions but pointers to similar patients these predictions are based on. As an additional requirement, the system must be adjustable by domain experts. We identified a setup using prototypical networks as suitable to fulfill these needs and presented an architecture that combines such networks with pre-trained language models.

Use of LLMs in interpretable systems. The analysis of model behavior we presented in Chapter 5 highlighted the importance of transparency when using deep learning models such as LLMs. Therefore, when adapting models to domain requirements, the interpretability of such models has a high priority. While LLMs are commonly used as black box models, our method *ProtoPatient* allows us to integrate them into an interpretable setup. We achieve this by using LLMs for encoding clinical notes into an interpretable prototypical network. An additional attention-based layer allows us to identify which tokens the model considers important for a decision. This way, it is possible to communicate faithful explanations of model predictions to domain experts while utilizing the favourable capabilities of LLMs, which we analyzed in the first part of this dissertation. The release of a demo application[1] helps to further refine model requirements, to understand its behavior, and to quickly identify strengths and weaknesses of the system.

9.2 Review of Research Questions

In Section 1 of this dissertation, we posed four research questions towards understanding the capabilities of large language models and adaptation strategies for specialized domains. In the following, we summarize our findings for each of these questions.

RQ1: What are common errors of machine learning models in specialized domains? How can large language models help to address them?

We conducted an ensemble-based error analysis to identify common mistakes of pre-LLM machine learning algorithms to understand main challenges of domain-specific NLP tasks. Our results showed three main sources of errors: 1. *Lack of contextual awareness.* The evaluated models often placed too much weight on single

[1]Demo application available at https://protopatient.demo.datexis.com.

tokens independent of their context. Transformer-based LLMs are focussed on the highly contextualized encoding of tokens due to their self-attention mechanism, in which each token has a 1-to-1 connection to all others tokens in the text. This allows to counteract the observed errors in many cases. 2. *Noisy training labels.* This problem occurs due to highly subjective annotation tasks with many influencing external factors, such as the identity of the annotator of a toxic language sample. The lack of an objective ground-truth is typical for specialized domains and was also observed when working with clinical data as described in Section 8.7. While LLMs can also be affected by this problem, their pre-training on large scale data makes them more robust against noisy labels as shown by Tänzer et al. [2022]. 3. *Missing world knowledge.* Due to data silos, labeled training data in specialized domains typically does not contain all knowledge required to solve a task. In our analysis, we found that missing world knowledge often leads to classification errors. LLMs can again benefit from their pre-training in this regard. The large corpora they are trained on contain many additional signals and help to complement missing world knowledge. With research question 2, we take a closer look at how such knowledge is stored within pre-trained large language models.

RQ2: How do large language models process information throughout their layers?

LLMs like BERT that are based on Transformer encoders are built of multiple layers of encoder blocks. In each block, the input sequence gets transformed into a new representation which is finally used as input to a classification head depending on the task to solve. This architecture combined with model pre-training has shown to work well on a multitude of NLP tasks indicating that information is processed in an effective way throughout the layers. To understand *how* these transformations work and which influence pre-training and fine-tuning has on this process, we conducted a layer-wise analysis of BERT-based models. We used models fine-tuned on Question Answering tasks, since such downstream tasks often require multiple steps of information processing. Our analysis revealed that the transformations take part in multiple phases. First, we see semantic clustering, followed by a phase in which entities, coreferences and their attributes are clustered. The last two phases are task-specific and first match the query with relevant parts of the input to then extract the answer tokens from the rest of the document tokens.

The first two phases are mostly task-independent and learned during pre-training. This allows to focus on domain- and task-specific transformations in the later layers, which are more strongly altered during fine-tuning. This way, the models are learning universal representations (mostly stored in their lower layers) during pre-training, which reduces the amount of required labeled training data for downstream tasks in the fine-tuning stage.

RQ3: How can we incorporate domain-specific knowledge into LLMs in the clinical domain?

We explored different ways to adapt general domain language models to the clinical domain. The most common method for incorporating domain-specific knowledge into LLMs is language model fine-tuning as defined by [Howard and Ruder, 2018]. We found that using training data that is closely related to the downstream task can be more beneficial than simply using larger amounts of data in this training state. Since such data is not always available, we investigated further ways to include clinical knowledge into language models. In that regard, we introduced a new domain- and task-specific pre-training objective (*clinical outcome pre-training*) and showed that it is an effective method to integrate further clinical knowledge into our models. The approach is inspired by the idea that clinical professionals learn both from experience and from literature. Simultaneously, we used public textual sources from clinical cases and from medical articles as input to the pre-training task and showed that the resulting models achieve better results than the baselines.

As another way to incorporate signals missing from training data, we identified the use of domain-specific resources, which do not necessarily have to be text-based. As one example, we have shown how the ICD (International Classification of Diseases) taxonomy can be used for model adaptation by incorporating hierarchical information about diseases into the model weights. Depending on the use case at hand, the type of resource that is beneficial to the task varies. However, in general, our results encourage to consider data sources beyond text to complement model weights with additional domain knowledge.

RQ4: How can we make large language models more transparent to serve domain requirements?

To answer this research question, we worked closely with domain experts, in our case, doctors with clinical experience. This cooperation allowed us to identify the most important requirements and multiple tasks within the clinical domain to address with neural NLP systems. An essential factor for such systems is the ability to effectively communicate model results to the domain experts. In the clinical domain, such communication includes both the explanation of individual model predictions, but also to highlight the general abilities and behavior of the applied models. We showed that both can be achieved while using black box models such as LLMs.

To obtain faithful model explanations, we presented an architecture that uses LLMs as building blocks within an interpretable system based on prototypical networks. The system is able to explain model decisions on a token-basis and to additionally point towards patients that are similar to the current case, fulfilling two important domain-specific requirements. For communicating the general behavior of multiple LLMs in the clinical setting, we introduced a behavioral testing framework that reveals model strengths but also potential adverse effects, such as learned biases.

While the requirements of specialized domains are manifold and diverse, transparency is a shared need in domain-specific setups. It allows domain experts to better judge and potentially improve the capabilities of the models.

Our approaches further demonstrate how LLMs can be integrated into different systems which can then be tailored to the requirements of individual domains.

9.3 Future Work

The presented work on exploration and adaptation of large language models provides many possibilities for continuation. In the following, we present two directions for future research that we consider particularly promising for specialized domains.

9.3.1 Integration of Multimodal and Multilingual Data

The approaches presented in this dissertation are limited to textual data in English. However, we see large potential for extending NLP systems for specialized domains to multiple languages and modalities. First, to serve a wider audience of domain experts, and second, to enable knowledge transfer between languages and regions around the world. In the following, we briefly discuss both strategies and point towards some promising approaches.

Multilingual domain-specific models. The majority of data collected in specialized domains, such as the clinical, is written in regional languages. The quality of domain-specific pre-trained language models for individual languages other than English varies depending on the amount of textual data available in this language and domain. In most cases, such data is rare, which impedes the availability of domain- and language-specific models. In Papaioannou et al. [2022], we explore how clinical LLMs pre-trained on English text can be beneficial for languages other than English in a sequential transfer learning setting. The results show that such knowledge transfer can improve the performance of models in lower resource languages. However, adding additional sources from different languages does not always lead to improvements. In times of global health events, such as the Covid-19 pandemic, the transfer of clinical knowledge around the globe becomes a pressing manner. In an increasingly globalized world, the same holds true for other specialized domains. That is why we see large potential in the development of multilingual and cross-lingual language models [Conneau and Lample, 2019]. Such models are already widely used for general domain text, but are less common for specialized domains. Promising approaches in this direction were proposed by Li et al. [2020], Jørgensen et al. [2021], and Verma et al. [2022], introducing strategies for domain adaptation of multilingual pre-trained models.

Integrating multiple modalities into LLMs. This dissertation considers text as the only input to large language models. However, data in specialized domains is usually comprised of multiple modalities. In the clinical domain, there exists an abundance of sensory and laboratory data, stored in tabular forms, but also image data, such as radiology scans. Simultaneously, determining whether an online user comment contains toxicity can be dependent on images or videos included in a posting. Consequently, we can only process the full information of a task if we use models that allow multimodal input. The last years have seen increasing progress in the direction of LLMs for multiple modalities, especially those combining vision and language models (e.g. CLIP [Radford et al., 2021] and VilBERT [Lu et al., 2019]). Those approaches are based on the idea of mapping input data from different modalities into a joint vector space, in which their semantic meaning is preserved, and combined. Ideally, we can train models to produce such semantic representations regardless of the input modality. Approaches in this direction include the Perceiver model [Jaegle et al., 2021], which uses the same underlying Transformer architecture for multiple modalities, and the recently published PaLM-E model [Driess et al., 2023]. Here, inputs are simply embedded into language sequences and then fed into a large language model able to encode combinations of modalities with a common language modeling objective. We expect similar approaches to play an increasing role in NLP for specialized domains. Acosta et al. [2022] give a comprehensive overview of the challenges and potentials of multimodal models for the biomedical domain. Especially domains with an inherent lack of data can benefit from such strategies that complement all available sources into a more complete input representation.

9.3.2 Efficient Methods for Large Language Models

One problem with the current development of LLMs is the ever-growing need for resources. These resources include data, computation, memory, and time. Since they all require significant investments, it becomes harder for companies or research groups with less funding to participate in state-of-the-art research. This does not only include training such models but also the inference and, in that way, the evaluation of models. However, when publicly available models can only be evaluated by certain parties, checking them for possible harms and thereby improving them steadily becomes increasingly difficult. That is why research towards efficient use of large language models becomes increasingly important.

Efficiency can be addressed across multiple dimensions, related to the affected resources listed above. It also covers all stages of the NLP pipeline, such as efficient use of data, model design, pre-training and fine-tuning of LLMs, inference, and hardware aspects. In Treviso et al. [2022b], we discuss existing methods for efficient use of LLMs across these different stages as visualized in Figure 9.1.

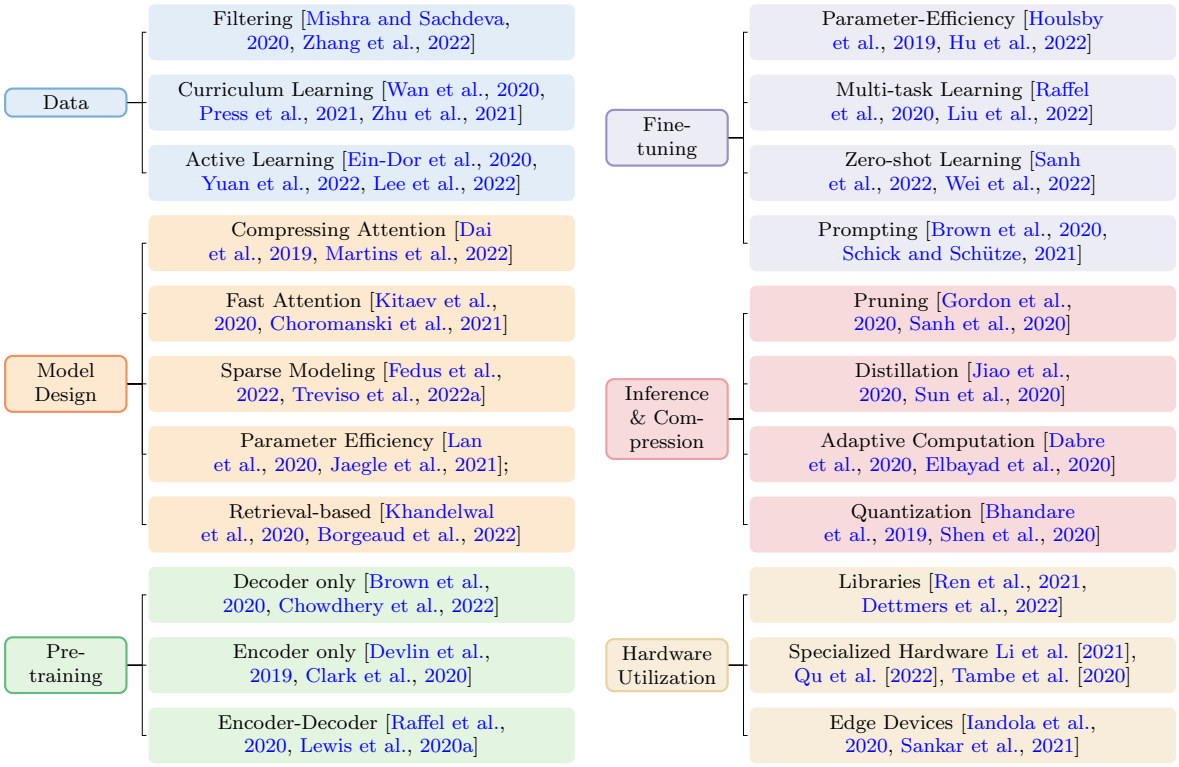


Figure 9.1. Typology of efficient NLP methods as surveyed by Treviso et al. [2022b].

Many of these methods can decrease resource usage significantly without impacting performance by large margins. However, we also identify challenges, such as trade-offs between different dimensions of efficiency and the lack of simple evaluation frameworks.

Considering the current direction of NLP research, which puts a focus on scaling, as discussed in 2.1.6, we especially recommend future research for LLMs in specialized domains to think efficiency-first. Most resources in specialized domains are naturally limited, and therefore, prioritizing efficiency across multiple dimensions will be particularly beneficial by enabling faster development, progress sharing, and, ultimately, safer applications.

10 Curriculum Vitae

Studies

since 2022	**PhD Studies** Gottfried Wilhelm Leibniz Universität Hannover	
	Informatik	Hannover, Germany
2014–2017	**Master of Science Ø1.1** Berliner Hochschule für Technik	
	Medieninformatik	Berlin, Germany
2010–2014	**Bachelor of Engineering Ø1.4** Hochschule der Medien Stuttgart	
	Audiovisuelle Medien	Stuttgart, Germany

Professional Experience

since 2023	**Applied Research Scientist** Grammarly Germany GmbH
	Berlin, Germany
2020–2023	**Data Science Consultant** self-employed
	Multiple locations
2018–2022	**Research Associate** Berliner Hochschule für Technik
	Berlin, Germany
2016–2017	**Software Developer** kreait GmbH
	Berlin, Germany
2014–2015	**Mobile Application Developer** Two Bulls GmbH
	Berlin, Germany

Publications

Please refer to the foreword of this dissertation for a full list of scientific publications.

Bibliography

A. B. Abacha and D. Demner-Fushman. A question-entailment approach to question answering. *BMC Bioinform.*, 20(1):511:1–511:23, 2019.

J. N. Acosta, G. J. Falcone, P. Rajpurkar, and E. J. Topol. Multimodal biomedical ai. *Nature Medicine*, 28(9):1773–1784, 2022.

E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jindi, T. Naumann, and M. McDermott. Publicly Available Clinical BERT Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June 2019. ACL.

M. Anderson, S. Moscou, C. Fulchon, and D. Neuspiel. The role of race in the clinical presentation. *Family medicine*, 33:430–4, 2001.

D. Araci. Finbert: Financial sentiment analysis with pre-trained language models. *CoRR*, abs/1908.10063, 2019.

S. Arnold, R. Schneider, P. Cudré-Mauroux, F. A. Gers, and A. Löser. SECTOR: A neural model for coherent topic segmentation and classification. *Trans. Assoc. Comput. Linguistics*, 7:169–184, 2019.

S. Arnold, B. van Aken, P. Grundmann, F. A. Gers, and A. Löser. Learning contextualized document representations for healthcare answer retrieval. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 1332–1343. ACM / IW3C2, 2020.

A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion*, 58:82–115, 2020.

P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3256–3274. Association for Computational Linguistics, 2020.

L. J. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

P. Badjatiya, S. Gupta, M. Gupta, and V. Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017*, pages 759–760. ACM, 2017.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

S. Barocas, K. Crawford, A. Shapiro, and H. Wallach. The problem with bias: Allocative versus representational harms in machine learning. In *SIGCIS, Philadelphia, PA.*, 2017.

T. Baumel, J. Nassour-Kassis, R. Cohen, M. Elhadad, and N. Elhadad. Multi-label classification of patient notes: Case study on ICD code assignment. In *The Workshops of the The Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, volume WS-18 of *AAAI Technical Report*, pages 409–416. AAAI Press, 2018.

B. Beizer. *Black-box testing - techniques for functional testing of software and systems.* Wiley, 1995. ISBN 978-0-471-12094-0.

Y. Belinkov, N. Durrani, F. Dalvi, H. Sajjad, and J. R. Glass. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 861–872. Association for Computational Linguistics, 2017.

E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, pages 610–623. ACM, 2021.

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, 2003.

A. Bhandare, V. Sripathi, D. Karkada, V. Menon, S. Choi, K. Datta, and V. A. Saletore. Efficient 8-bit quantization of transformer neural machine language translation model. *CoRR*, abs/1906.00532, 2019.

P. Bhatia, B. Celikkaya, and M. Khalilia. Joint entity extraction and assertion detection for clinical text. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 954–959. Association for Computational Linguistics, 2019.

P. Blinov, M. Avetisian, V. Kokh, D. Umerenkov, and A. Tuzhilin. Predicting clinical diagnosis from patients electronic health records using bert-based neural networks. In *Artificial Intelligence in Medicine - 18th International Conference on Artificial Intelligence in Medicine, AIME 2020, Minneapolis, MN, USA, August 25-28, 2020, Proceedings*, volume 12299 of *Lecture Notes in Computer Science*, pages 111–121. Springer, 2020.

W. Boag, D. Doss, T. Naumann, and P. Szolovits. What's in a Note? Unpacking Predictive Value in Clinical Note Representations. *AMIA Summits on Translational Science Proceedings*, 2018:26 – 34, 2018.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics*, 5:135–146, 2017.

S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. van den Driessche, J. Lespiau, B. Damoc, A. Clark, D. de Las Casas, A. Guy, J. Menick, R. Ring, T. Hennigan, S. Huang, L. Maggiore, C. Jones, A. Cassirer, A. Brock, M. Paganini, G. Irving, O. Vinyals, S. Osindero, K. Simonyan, J. W. Rae, E. Elsen, and L. Sifre. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR, 2022.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

R. A. Bulatao and N. B. Anderson. Understanding racial and ethnic differences in health in late life: A research agenda. *National Academies Press (US)*, 2004.

P. Burnap and M. L. Williams. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. volume 7, pages 223–242, 2015.

P. Burnap and M. L. Williams. Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data Sci.*, 5(1):11, 2016.

I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos. LEGAL-BERT: the muppets straight out of law school. *CoRR*, abs/2010.02559, 2020.

W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *J. Biomed. Informatics*, 34(5):301–310, 2001.

C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. Su. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8928–8939, 2019.

L. Chen. Attention-based Deep Learning System for Negation and Assertion Detection in Clinical Notes. *International Journal of Artificial Intelligence and Applications*, 10(1), 2019.

Y. Chen, Y. Zhou, S. Zhu, and H. Xu. Detecting offensive language in social media to protect adolescent online safety. pages 71–80, 2012.

K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL, 2014.

E. Choi, S. Biswal, B. A. Malin, J. Duke, W. F. Stewart, and J. Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Proceedings of the Machine Learning for Health Care Conference, MLHC 2017, Boston, Massachusetts, USA, 18-19 August 2017*, volume 68 of *Proceedings of Machine Learning Research*, pages 286–305. PMLR, 2017.

E. Choi, C. Xiao, W. F. Stewart, and J. Sun. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 4552–4562, 2018.

K. M. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlós, P. Hawkins, J. Q. Davis, A. Mohiuddin, L. Kaiser, D. B. Belanger, L. J. Colwell, and A. Weller. Rethinking attention with performers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311, 2022.

K. Clark, M. Luong, Q. V. Le, and C. D. Manning. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, 2011.

P. Comon. Independent component analysis, A new concept? *Signal Process.*, 36(3): 287–314, 1994.

A. Conneau and D. Kiela. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA), 2018.

A. Conneau and G. Lample. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067, 2019.

D. R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

R. Dabre, R. Rubino, and A. Fujita. Balancing cost and benefit with tied-multi transformers. In *Proceedings of the Fourth Workshop on Neural Generation and Translation, NGT@ACL 2020, Online, July 5-10, 2020*, pages 24–34. Association for Computational Linguistics, 2020.

M. Dadvar, D. Trieschnigg, R. Ordelman, and F. de Jong. Improving cyberbullying detection with user context. In *Advances in Information Retrieval - 35th European Conference on IR Research, ECIR 2013, Moscow, Russia, March 24-27, 2013. Proceedings*, volume 7814 of *Lecture Notes in Computer Science*, pages 693–696. Springer, 2013.

Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988. Association for Computational Linguistics, 2019.

M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen. A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pages 447–459. Association for Computational Linguistics, 2020.

A. Das, C. Gupta, V. Kovatchev, M. Lease, and J. J. Li. Prototex: Explaining model decisions with prototype tensors. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2986–2997. Association for Computational Linguistics, 2022.

T. Davidson, D. Warmsley, M. W. Macy, and I. Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017*, pages 512–515. AAAI Press, 2017.

B. de Bruijn, C. Cherry, S. Kiritchenko, J. D. Martin, and X. Zhu. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *J. Am. Medical Informatics Assoc.*, 18(5):557–562, 2011.

S. Deng, N. Zhang, J. Kang, Y. Zhang, W. Zhang, and H. Chen. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 151–159. ACM, 2020.

T. Dettmers, M. Lewis, S. Shleifer, and L. Zettlemoyer. 8-bit optimizers via block-wise quantization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

J. Devlin, M. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

K. Dinakar, B. Jones, C. Havasi, H. Lieberman, and R. W. Picard. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Trans. Interact. Intell. Syst.*, 2(3):18:1–18:30, 2012.

D. Dligach, M. Afshar, and T. A. Miller. Toward a clinical text encoder: pretraining for clinical natural language processing with applications to substance misuse. *J. Am. Medical Informatics Assoc.*, 26(11):1272–1278, 2019.

J. Dodge, I. Gurevych, R. Schwartz, E. Strubell, and B. van Aken. Efficient and equitable natural language processing in the age of deep learning (dagstuhl seminar 22232). *Dagstuhl Reports*, 12(6):14–27, 2022.

H. Dong, V. Suárez-Paniagua, W. Whiteley, and H. Wu. Explainable automated coding of clinical notes using hierarchical label-wise attention networks and label embedding initialisation. *J. Biomed. Informatics*, 116:103728, 2021.

F. K. Dosilovic, M. Brcic, and N. Hlupic. Explainable artificial intelligence: A survey. In *41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018, Opatija, Croatia, May 21-25, 2018*, pages 210–215. IEEE, 2018.

D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence. Palm-e: An embodied multimodal language model. *CoRR*, abs/2303.03378, 2023.

L. Ein-Dor, A. Halfon, A. Gera, E. Shnarch, L. Dankin, L. Choshen, M. Danilevsky, R. Aharonov, Y. Katz, and N. Slonim. Active learning for BERT: an empirical study.

In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7949–7962. Association for Computational Linguistics, 2020.

M. Elbayad, J. Gu, E. Grave, and M. Auli. Depth-adaptive transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

N. Fakhraie. *What's in a Note? Sentiment Analysis in Online Educational Forums*. University of Toronto (Canada), 2011.

M. Falis, M. Pajak, A. Lisowska, P. Schrempf, L. Deckers, S. Mikhael, S. A. Tsaftaris, and A. O'Neil. Ontological attention ensembles for capturing semantic concepts in ICD code prediction from clinical text. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis LOUHI@EMNLP 2019, Hong Kong, November 3, 2019*, pages 168–177. Association for Computational Linguistics, 2019.

W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39, 2022.

J. Feng, Q. Wei, and J. Cui. Prototypical networks relation classification model based on entity convolution. *Comput. Speech Lang.*, 77:101432, 2023.

J. R. Firth. A synopsis of linguistic theory, 1930-1955. WORD, 1957.

A. Flores, J. Herman, G. Gates, and T. Brown. How many adults identify as transgender in the united states? *Los Angeles, CA: The Williams Institute*, 2016.

J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232, 2001.

K. P. F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2, 1901.

B. Gambäck and U. K. Sikdar. Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online, ALW@ACL 2017, Vancouver, BC, Canada, August 4, 2017*, pages 85–90. Association for Computational Linguistics, 2017.

L. Gao and R. Huang. Detecting online hate speech using context aware models. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, Varna, Bulgaria, September 2 - 8, 2017*, pages 260–266. INCOMA Ltd., 2017.

B. A. Garner. Black's law dictionary, 2nd ed. St. Paul, MN, 2014. Thomson Reuters.

S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos. Convolutional neural networks for toxic comment classification. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence, SETN 2018, Patras, Greece, July 09-12, 2018*, pages 35:1–35:6. ACM, 2018.

M. Ghassemi, T. Naumann, F. Doshi-Velez, N. Brimmer, R. Joshi, A. Rumshisky, and P. Szolovits. Unfolding physiological state: mortality modelling in intensive care units. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 75–84. ACM, 2014.

A. Goddu, K. O'Conor, S. Lanzkron, M. Saheed, S. Saha, C. Haywood, and M. C. Beach. Do words matter? stigmatizing language and the transmission of bias in the medical record. *Journal of General Internal Medicine*, 33, 2018.

J. Golbeck, Z. Ashktorab, R. O. Banjo, A. Berlinger, S. Bhagwan, C. Buntain, P. Cheakalos, A. A. Geller, Q. Gergory, R. K. Gnanasekaran, R. R. Gunasekaran, K. M. Hoffman, J. Hottle, V. Jienjitlert, S. Khare, R. Lau, M. J. Martindale, S. Naik, H. L. Nixon, P. Ramachandran, K. M. Rogers, L. Rogers, M. S. Sarin, G. Shahane, J. Thanki, P. Vengataraman, Z. Wan, and D. M. Wu. A large labeled corpus for online harassment research. In *Proceedings of the 2017 ACM on Web Science Conference, WebSci 2017, Troy, NY, USA, June 25 - 28, 2017*, pages 229–233. ACM, 2017.

Y. Goldberg. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287, 2019.

M. A. Gordon, K. Duh, and N. Andrews. Compressing BERT: studying the effects of weight pruning on transfer learning. In *Proceedings of the 5th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2020, Online, July 9, 2020*, pages 143–155. Association for Computational Linguistics, 2020.

A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385 of *Studies in Computational Intelligence*. Springer, 2012. ISBN 978-3-642-24796-5.

E. Greevy and A. F. Smeaton. Classifying racist texts using a support vector machine. In *SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004*, pages 468–469. ACM, 2004.

Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Heal.*, 3(1):2:1–2:23, 2022.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5):93:1–93:42, 2019.

S. Gururangan, A. Marasovic, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8342–8360. Association for Computational Linguistics, 2020.

H. Harkema, J. N. Dowling, T. Thornblade, and W. W. Chapman. Context: An algorithm for determining negation, experiencer, and temporal status from clinical reports. *J. Biomed. Informatics*, 42(5):839–851, 2009.

Z. S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1954.

M. Hashir and R. Sawhney. Towards unstructured mortality prediction with free-text clinical notes. *J. Biomed. Informatics*, 108:103489, 2020.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

C. V. Hee, E. Lefever, B. Verhoeven, J. Mennes, B. Desmet, G. D. Pauw, W. Daelemans, and V. Hoste. Detection and fine-grained classification of cyberbullying events. In *Recent Advances in Natural Language Processing, RANLP 2015, 7-9 September, 2015, Hissar, Bulgaria*, pages 672–680. RANLP 2015 Organising Committee / ACL, 2015.

B. Heinzerling. Nlp's clever hans moment has arrived. *The Journal of Cognitive Science*, 21:161–170, 2020.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. *A Field Guide to Dynamical Recurrent Neural Networks*, 2001.

N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 2019.

J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 328–339. Association for Computational Linguistics, 2018.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

K. Huang, J. Altosaar, and R. Ranganath. ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. In *Proceedings of ACM Conference on Health, Inference, and Learning, CHIL 2020*, Online, 2019. ACM.

F. N. Iandola, A. E. Shaw, R. Krishna, and K. Keutzer. Squeezebert: What can computer vision teach NLP about efficient neural networks? In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing, SustaiNLP@EMNLP 2020, Online, November 20, 2020*, pages 124–135. Association for Computational Linguistics, 2020.

A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR, 2021.

S. Jain and B. C. Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3543–3556. Association for Computational Linguistics, 2019.

S. Jain, R. Mohammadi, and B. C. Wallace. An analysis of attention over clinical notes for predictive tasks. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 15–21, Minneapolis, Minnesota, USA, 2019. Association for Computational Linguistics.

A. Jha and R. Mamidi. When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. In *Proceedings of the Second Workshop on NLP and Computational Social Science, NLP+CSS@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 7–16. Association for Computational Linguistics, 2017.

X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu. Tinybert: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4163–4174. Association for Computational Linguistics, 2020.

Y. Jo, L. Lee, and S. Palaskar. Combining LSTM and latent topic modeling for mortality prediction. *CoRR*, abs/1709.02842, 2017.

A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. MIMIC-III, a Freely Accessible Critical Care Database. *Scientific Data*, 3(1):1–9, 2016.

K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, 60(5):493–502, 2004.

R. K. Jørgensen, M. Hartmann, X. Dai, and D. Elliott. mdapt: Multilingual domain adaptive pretraining in a single model. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3404–3418. Association for Computational Linguistics, 2021.

G. Kennedy, A. McCollough, E. Dixon, A. Bastidas, J. Ryan, C. Loo, and S. Sahay. Technology solutions to combat online harassment. In *Proceedings of the First Workshop on Abusive Language Online, ALW@ACL 2017, Vancouver, BC, Canada, August 4, 2017*, pages 73–77. Association for Computational Linguistics, 2017.

S. Khadanga, K. Aggarwal, S. R. Joty, and J. Srivastava. Using clinical notes with time series data for ICU management. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6431–6436. Association for Computational Linguistics, 2019.

A. Khandelwal and S. Sawant. Negbert: A transfer learning approach for negation detection and scope resolution. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 5739–5748. European Language Resources Association, 2020.

U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

O. Khosravizadeh, S. Vatankhah, P. Bastani, R. Kalhor, S. Alirezaei, and F. Doosty. Factors affecting length of stay in teaching hospitals of a middle-income country. *Electronic physician*, 8(10):3042–3047, 2016.

Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL, 2014.

P. Kindermans, K. Schütt, K. Müller, and S. Dähne. Investigating the influence of noise and distractors on the interpretation of neural networks. *CoRR*, abs/1611.07270, 2016.

N. Kitaev, L. Kaiser, and A. Levskaya. Reformer: The efficient transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. Large language models are zero-shot reasoners. In *NeurIPS*, 2022.

I. Kwok and Y. Wang. Locate the hate: Detecting tweets against blacks. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA*. AAAI Press, 2013.

Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*, 36(4): 1234–1240, 2020.

J. Lee, J. Klie, and I. Gurevych. Annotation curricula to implicitly train non-expert annotators. *Comput. Linguistics*, 48(2):343–373, 2022.

J. Lee, H. Puerto, B. van Aken, Y. Arase, J. Z. Forde, L. Derczynski, A. Rücklé, I. Gurevych, R. Schwartz, E. Strubell, and J. Dodge. Surveying (dis)parities and concerns of compute hungry NLP research. *CoRR*, abs/2306.16900, 2023.

D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics, 2020a.

P. S. H. Lewis, M. Ott, J. Du, and V. Stoyanov. Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop, ClinicalNLP@EMNLP 2020, Online, November 19, 2020*, pages 146–157. Association for Computational Linguistics, 2020b.

J. Li, W. Monroe, and D. Jurafsky. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220, 2016.

J. Li, R. He, H. Ye, H. T. Ng, L. Bing, and R. Yan. Unsupervised domain adaptation of a pretrained cross-lingual language model. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3672–3678. ijcai.org, 2020.

Q. Li, X. Zhang, J. Xiong, W. Hwu, and D. Chen. Efficient methods for mapping neural machine translator on fpgas. *IEEE Trans. Parallel Distributed Syst.*, 32(7):1866–1877, 2021.

X. Li and D. Roth. Learning question classifiers. In *19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, August 24 - September 1, 2002*, 2002.

Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Z. C. Lipton. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43, 2018.

D. Liu, D. Dligach, and T. A. Miller. Two-stage federated phenotyping and patient representation learning. In *Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 283–291. Association for Computational Linguistics, 2019a.

H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *NeurIPS*, 2022.

J. Liu, Z. Zhang, and N. Razavian. Deep EHR: chronic disease prediction using medical notes. In *Proceedings of the Machine Learning for Healthcare Conference, MLHC 2018, 17-18 August 2018, Palo Alto, California*, volume 85 of *Proceedings of Machine Learning Research*, pages 440–464. PMLR, 2018.

N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1073–1094. Association for Computational Linguistics, 2019b.

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.*, 55(9):195:1–195:35, 2023.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019c.

S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2):129–136, 1982. doi: 10.1109/TIT.1982.1056489.

C. Logé, E. Ross, D. Y. A. Dadey, S. Jain, A. Saporta, A. Y. Ng, and P. Rajpurkar. Q-pain: A question answering dataset to measure social bias in pain management. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021.

I. Loshchilov and F. Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017.

J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13–23, 2019.

P. H. Martins, Z. Marinho, and A. F. T. Martins. ∞-former: Infinite memory transformer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5468–5485. Association for Computational Linguistics, 2022.

B. McCann, N. S. Keskar, C. Xiong, and R. Socher. The natural language decathlon: Multitask learning as question answering. *CoRR*, abs/1806.08730, 2018.

M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989.

Y. Mehdad and J. R. Tetreault. Do characters abuse more than words? In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, pages 299–303. The Association for Computer Linguistics, 2016.

O. Melamud, J. Goldberger, and I. Dagan. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 51–61. ACL, 2016.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

Y. Ming, P. Xu, H. Qu, and L. Ren. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 903–913. ACM, 2019.

S. Mishra and B. S. Sachdeva. Do we need to create big datasets to learn a task? In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing, SustaiNLP@EMNLP 2020, Online, November 20, 2020*, pages 169–173. Association for Computational Linguistics, 2020.

M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*, pages 220–229. ACM, 2019.

S. Moscou, M. R. Anderson, J. B. Kaplan, and L. Valencia. Validity of racial/ethnic classifications in medical records data: an exploratory study. *American journal of public health*, 93(7):1084–1086, 2003.

J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, and J. Eisenstein. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1101–1111. Association for Computational Linguistics, 2018.

A. Nelson. Unequal treatment: confronting racial and ethnic disparities in health care. *Journal of the national medical association*, 94(8):666, 2002.

D. Njagi, Z. Zuping, D. Hanyurwimfura, and J. Long. A lexicon-based approach for hate speech detection. In *International Journal of Multimedia and Ubiquitous Engineering*, volume 10, pages 215–230, 2015.

C. Nobata, J. R. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 145–153. ACM, 2016.

M. Oleynik, A. Kugic, Z. Kasác, and M. Kreuzthaler. Evaluating shallow and deep learning strategies for the 2018 n2c2 shared task on clinical text classification. *J. Am. Medical Informatics Assoc.*, 26(11):1247–1254, 2019.

OpenAI. Introducing chatgpt. https://openai.com/blog/chatgpt, 2022. Accessed: 2023-03-18.

S. Paasch-Colberg, B. van Aken, S. Christian, L. Laugwitz, A. Löser, J. Trebbe, and M. Emmer. Digging deeper: Extending the error analysis of a hate speech algorithm with information rich data. In *6th International Conference on Computational Social Science (IC2S2) Amherst St. Cambridge*, 2020.

J. Papaioannou, P. Grundmann, B. van Aken, A. Samaras, I. Kyparissidis, G. Giannakoulas, F. A. Gers, and A. Löser. Cross-lingual knowledge transfer for clinical phenotyping. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 900–909. European Language Resources Association, 2022.

A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2249–2255. The Association for Computational Linguistics, 2016.

J. H. Park and P. Fung. One-step and two-step classification for abusive language detection on twitter. pages 41–45, 2017.

J. Pavlopoulos, P. Malakasiotis, and I. Androutsopoulos. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1125–1135. Association for Computational Linguistics, 2017.

X. Peng, G. Long, T. Shen, S. Wang, and J. Jiang. Self-attention enhanced patient journey understanding in healthcare system. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020, Proceedings, Part III*, volume 12459 of *Lecture Notes in Computer Science*, pages 719–735. Springer, 2020.

Y. Peng, S. Yan, and Z. Lu. Transfer learning in biomedical natural language processing: An evaluation of BERT and elmo on ten benchmarking datasets. In *Proceedings of the*

18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, August 1, 2019, pages 58–65. Association for Computational Linguistics, 2019.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL, 2014.

S. Perera, A. P. Sheth, K. Thirunarayan, S. Nair, and N. Shah. Challenges in understanding clinical notes: why NLP engines fall short and where background knowledge can help. In *Proceedings of the 2013 International Workshop on Data Management & Analytics for Healthcare, DARE 2013, San Francisco, California, USA, November 1, 2013*, pages 21–26. ACM, 2013.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018.

E. R. Pfaff, M. Crosskey, K. Morton, and A. Krishnamurthy. Clinical annotation research kit (clark): Computable phenotyping using machine learning. *JMIR medical informatics*, 8(1):e16042, 2020.

O. Press, N. A. Smith, and M. Lewis. Shortformer: Better language modeling using shorter inputs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5493–5505. Association for Computational Linguistics, 2021.

M. Ptaszynski, J. K. K. Eronen, and F. Masui. Learning deep on cyberbullying is always better than brute force. In *Proceedings of the Linguistic and Cognitive Approaches To Dialog Agents Workshop co-located with the 26th International Joint Conference on Artificial Intelligence, LaCATODA@IJCAI 2017, Melbourne, Australia, August 21, 2017*, volume 1926 of *CEUR Workshop Proceedings*, pages 3–10. CEUR-WS.org, 2017.

R. Puri and B. Catanzaro. Zero-shot text classification with generative language models. *CoRR*, abs/1912.10165, 2019.

J. Qian, M. ElSherief, E. M. Belding, and W. Y. Wang. Leveraging intra-user and inter-user representation learning for automated hate speech detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 118–123. Association for Computational Linguistics, 2018.

Z. Qian, P. Li, Q. Zhu, G. Zhou, Z. Luo, and W. Luo. Speculation and negation scope detection via convolutional neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 815–825. The Association for Computational Linguistics, 2016.

Y. Qiao, C. Xiong, Z. Liu, and Z. Liu. Understanding the behaviors of BERT in ranking. *CoRR*, abs/1904.07531, 2019a.

Z. Qiao, X. Wu, S. Ge, and W. Fan. MNN: multimodal attentional neural networks for diagnosis prediction. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5937–5943. ijcai.org, 2019b.

X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang. Pre-trained Models for Natural Language Processing: A Survey. *Science China Technological Sciences*, 63:1872 – 1897, 2020. ISSN 1869-1900.

Z. Qu, L. Liu, F. Tu, Z. Chen, Y. Ding, and Y. Xie. DOTA: detect and omit weak attentions for scalable transformer acceleration. In *ASPLOS '22: 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Lausanne, Switzerland, 28 February 2022 - 4 March 2022*, pages 14–26. ACM, 2022.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics, 2016.

P. Rajpurkar, R. Jia, and P. Liang. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 784–789. Association for Computational Linguistics, 2018.

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 Digit. Medicine*, 4, 2021.

H. Ren, Y. Cai, X. Chen, G. Wang, and Q. Li. A two-phase prototypical network model for incremental few-shot relation classification. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 1618–1629. International Committee on Computational Linguistics, 2020.

J. Ren, S. Rajbhandari, R. Y. Aminabadi, O. Ruwase, S. Yang, M. Zhang, D. Li, and Y. He. Zero-offload: Democratizing billion-scale model training. In *2021 USENIX Annual Technical Conference, USENIX ATC 2021, July 14-16, 2021*, pages 551–564. USENIX Association, 2021.

M. T. Ribeiro, S. Singh, and C. Guestrin. Model-agnostic interpretability of machine learning. *CoRR*, abs/1606.05386, 2016.

M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. Beyond accuracy: Behavioral testing of NLP models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4902–4912. Association for Computational Linguistics, 2020.

W. J. Riley. Health disparities: gaps in access, quality and affordability of medical care. *Transactions of the American Clinical and Climatological Association*, 123:167, 2012.

J. Risch and R. Krestel. Aggression identification using deep learning and data augmentation. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018, Santa Fe, New Mexico, USA, August 25, 2018*, pages 150–158. Association for Computational Linguistics, 2018.

D. Robinson, Z. Zhang, and J. A. Tepper. Hate speech detection on twitter: Feature engineering v.s. feature selection. In *The Semantic Web: ESWC 2018 Satellite Events - ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers*, volume 11155 of *Lecture Notes in Computer Science*, pages 46–49. Springer, 2018.

A. Rogers, O. Kovaleva, and A. Rumshisky. A primer in bertology: What we know about how BERT works. *Trans. Assoc. Comput. Linguistics*, 8:842–866, 2020.

S. Rosenthal, K. Barker, and Z. Liang. Leveraging medical literature for section prediction in electronic health records. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4863–4872. Association for Computational Linguistics, 2019.

B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, and M. Wojatzki. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *CoRR*, abs/1701.08118, 2017.

C. Rosset. Turing-nlg: A 17-billion-parameter language model by microsoft. https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft, 02 2020. Accessed: 2023-03-18.

P. Röttger, B. Vidgen, D. Nguyen, Z. Waseem, H. Z. Margetts, and J. B. Pierrehumbert. Hatecheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 41–58. Association for Computational Linguistics, 2021.

S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2, 2019, Tutorial Abstracts*, pages 15–18. Association for Computational Linguistics, 2019.

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.*, 1(5):206–215, 2019.

M. Sahlgren. The distributional hypothesis. *The Italian Journal of Linguistics*, 20:33–54, 2008.

N. S. Samghabadi, S. Maharjan, A. P. Sprague, R. Diaz-Sprague, and T. Solorio. Detecting nastiness in social media. In *Proceedings of the First Workshop on Abusive Language Online, ALW@ACL 2017, Vancouver, BC, Canada, August 4, 2017*, pages 63–72. Association for Computational Linguistics, 2017.

V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.

V. Sanh, T. Wolf, and A. M. Rush. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. V. Nayak, D. Datta, J. Chang, M. T. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Févry, J. A. Fries, R. Teehan, T. L. Scao, S. Biderman, L. Gao, T. Wolf, and A. M. Rush. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

C. Sankar, S. Ravi, and Z. Kozareva. Proformer: Towards on-device LSH projection based transformers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2823–2828. Association for Computational Linguistics, 2021.

T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilic, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, J. Tow, A. M. Rush, S. Biderman, A. Webson, P. S. Ammanamanchi, T. Wang, B. Sagot, N. Muennighoff, A. V. del Moral, O. Ruwase, R. Bawden, S. Bekman,

A. McMillan-Major, I. Beltagy, H. Nguyen, L. Saulnier, S. Tan, P. O. Suarez, V. Sanh, H. Laurençon, Y. Jernite, J. Launay, M. Mitchell, C. Raffel, A. Gokaslan, A. Simhi, A. Soroa, A. F. Aji, A. Alfassy, A. Rogers, A. K. Nitzav, C. Xu, C. Mou, C. Emezue, C. Klamm, C. Leong, D. van Strien, D. I. Adelani, and et al. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100, 2022.

T. Schick and H. Schütze. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2339–2352. Association for Computational Linguistics, 2021.

A. Schmidt and M. Wiegand. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, SocialNLP@EACL 2017, Valencia, Spain, April 3, 2017*, pages 1–10. Association for Computational Linguistics, 2017.

R. Schneider, T. Oberhauser, P. Grundmann, F. A. Gers, A. Löser, and S. Staab. Is language modeling enough? evaluating effective embedding combinations. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4739–4748. European Language Resources Association, 2020.

M. Schuster and K. Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*, pages 5149–5152. IEEE, 2012.

R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni. Green AI. *Commun. ACM*, 63(12): 54–63, 2020.

T. Searle, Z. M. Ibrahim, and R. J. B. Dobson. Experimental evaluation and development of a silver-standard for the MIMIC-III clinical coding dataset. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing, BioNLP 2020, Online, July 9, 2020*, pages 76–85. Association for Computational Linguistics, 2020.

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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

E. Sergeeva, H. Zhu, P. Prinsen, and A. Tahmasebi. Negation Scope Detection in Clinical Notes and Scientific Abstracts: A Feature-enriched LSTM-based Approach. *AMIA Summits on Translational Science Proceedings*, 2019:212, 2019.

F. Shamout, T. Zhu, and D. A. Clifton. Machine learning for clinical outcome prediction. *IEEE reviews in Biomedical Engineering*, 14:116–126, 2020.

C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3): 379–423, 1948.

S. Sharma, S. Agrawal, and M. Shrivastava. Degree based classification of harmful speech using twitter data. pages 106–112, 2018.

S. Shen, Z. Dong, J. Ye, L. Ma, Z. Yao, A. Gholami, M. W. Mahoney, and K. Keutzer. Q-BERT: hessian based ultra low precision quantization of BERT. pages 8815–8821, 2020.

X. Shi, I. Padhi, and K. Knight. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1526–1534. The Association for Computational Linguistics, 2016.

M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *CoRR*, abs/1909.08053, 2019.

Y. Si and K. Roberts. Deep Patient Representation of Clinical Notes via Multi-Task Learning for Mortality Prediction. *AMIA Summits on Translational Science Proceedings*, 2019: 779–788, 2019.

J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4077–4087, 2017.

S. Snipes, S. Sellers, A. Tafawa, L. Cooper, J. Fields, and V. Bonham. Is race medically relevant? a qualitative study of physicians' attitudes about the role of race in treatment decision-making. *BMC health services research*, 11:183, 2011.

J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. Striving for simplicity: The all convolutional net. 2015.

A. L. Stangl, V. A. Earnshaw, C. H. Logie, W. van Brakel, L. C. Simbayi, I. Barré, and J. F. Dovidio. The health stigma and discrimination framework: a global, crosscutting framework to inform research, intervention development, and policy on health-related stigmas. *BMC medicine*, 17(1):1–13, 2019.

I. Straw. The automation of bias in medical artificial intelligence (AI): decoding the past to create a better future. *Artif. Intell. Medicine*, 110:101965, 2020.

E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3645–3650. Association for Computational Linguistics, 2019.

A. Stubbs and Ö. Uzuner. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus. *J. Biomed. Informatics*, 58:S20–S29, 2015.

A. Stubbs, C. Kotfila, and Ö. Uzuner. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1. *J. Biomed. Informatics*, 58:S11–S19, 2015.

S. Sun, Q. Sun, K. Zhou, and T. Lv. Hierarchical attention prototypical networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 476–485. Association for Computational Linguistics, 2019a.

T. Sun, A. Gaut, S. Tang, Y. Huang, M. ElSherief, J. Zhao, D. Mirza, E. M. Belding, K. Chang, and W. Y. Wang. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1630–1640. Association for Computational Linguistics, 2019b.

W. Sun, A. Rumshisky, and Ö. Uzuner. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *J. Am. Medical Informatics Assoc.*, 20(5):806–813, 2013a.

W. Sun, A. Rumshisky, and Ö. Uzuner. Annotating temporal information in clinical narratives. *J. Biomed. Informatics*, 46(6):S5–S12, 2013b.

Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou. Mobilebert: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2158–2170. Association for Computational Linguistics, 2020.

H. Suresh, J. J. Gong, and J. V. Guttag. Learning tasks for multitask learning: Heterogenous patient populations in the ICU. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 802–810. ACM, 2018.

M. Sushil, S. Suster, K. Luyckx, and W. Daelemans. Patient representation learning and interpretable evaluation using clinical notes. *J. Biomed. Informatics*, 84:103–113, 2018.

T. Tambe, C. Hooper, L. Pentecost, E. Yang, M. Donato, V. Sanh, A. M. Rush, D. Brooks, and G. Wei. Edgebert: Optimizing on-chip inference for multi-task NLP. *CoRR*, abs/2011.14203, 2020.

C. Tan and M. Chlebicki. Urinary tract infections in adults. *Singapore Medical Journal*, 57:485–490, 2016.

M. Tänzer, S. Ruder, and M. Rei. Memorisation versus generalisation in pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7564–7578. Association for Computational Linguistics, 2022.

Y. Tao, B. Godefroy, G. Genthial, and C. Potts. Effective feature representation for clinical text concept extraction. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 1–14, Minneapolis, Minnesota, USA, 2019. Association for Computational Linguistics.

I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. V. Durme, S. R. Bowman, D. Das, and E. Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

R. Tinn, H. Cheng, Y. Gu, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon. Fine-tuning large neural language models for biomedical natural language processing. *CoRR*, abs/2112.07869, 2021.

M. V. Treviso, A. Góis, P. Fernandes, E. R. Fonseca, and A. F. T. Martins. Predicting attention sparsity in transformers. In *Proceedings of the Sixth Workshop on Structured Prediction for NLP, SPNLP@ACL 2022, Dublin, Ireland, May 27, 2022*, pages 67–81. Association for Computational Linguistics, 2022a.

M. V. Treviso, T. Ji, J. Lee, B. van Aken, Q. Cao, M. R. Ciosici, M. Hassid, K. Heafield, S. Hooker, P. H. Martins, A. F. T. Martins, P. A. Milder, C. Raffel, E. Simpson, N. Slonim, N. Balasubramanian, L. Derczynski, and R. Schwartz. Efficient methods for natural language processing: A survey. *CoRR*, abs/2209.00099, 2022b.

P. D. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *J. Artif. Intell. Res.*, 37:141–188, 2010. doi: 10.1613/jair.2934.

Ö. Uzuner, Y. Luo, and P. Szolovits. Viewpoint paper: Evaluating the state-of-the-art in automatic de-identification. *J. Am. Medical Informatics Assoc.*, 14(5):550–563, 2007.

Ö. Uzuner, I. Goldstein, Y. Luo, and I. S. Kohane. Viewpoint paper: Identifying patient smoking status from medical discharge records. *J. Am. Medical Informatics Assoc.*, 15 (1):14–24, 2008.

Ö. Uzuner, I. Solti, and E. Cadag. Extracting medication information from clinical text. *J. Am. Medical Informatics Assoc.*, 17(5):514–518, 2010a.

Ö. Uzuner, I. Solti, F. Xia, and E. Cadag. Community annotation experiment for ground truth generation for the i2b2 medication challenge. *J. Am. Medical Informatics Assoc.*, 17(5):519–523, 2010b.

Ö. Uzuner, B. R. South, S. Shen, and S. L. DuVall. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *J. Am. Medical Informatics Assoc.*, 18(5):552–556, 2011.

Ö. Uzuner, A. Bodnari, S. Shen, T. Forbush, J. Pestian, and B. R. South. Evaluating the state of the art in coreference resolution for electronic medical records. *J. Am. Medical Informatics Assoc.*, 19(5):786–791, 2012.

B. van Aken, J. Risch, R. Krestel, and A. Löser. Challenges for toxic comment classification: An in-depth error analysis. In *Proceedings of the 2nd Workshop on Abusive Language Online, ALW@EMNLP 2018, Brussels, Belgium, October 31, 2018*, pages 33–42. Association for Computational Linguistics, 2018.

B. van Aken, B. Winter, A. Löser, and F. A. Gers. How does BERT answer questions?: A layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 1823–1832. ACM, 2019.

B. van Aken, B. Winter, A. Löser, and F. A. Gers. Visbert: Hidden-state visualizations for transformers. In *Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 207–211. ACM / IW3C2, 2020.

B. van Aken, J. Papaioannou, M. Mayrdorfer, K. Budde, F. A. Gers, and A. Löser. Clinical outcome prediction from admission notes using self-supervised knowledge integration. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 881–893. Association for Computational Linguistics, 2021a.

B. van Aken, I. Trajanovska, A. Siu, M. Mayrdorfer, K. Budde, and A. Löser. Assertion detection in clinical notes: Medical language models to the rescue? In *Proceedings of the Second Workshop on Natural Language Processing for Medical Conversations*, pages 35–40, 2021b.

B. van Aken, S. Herrmann, and A. Löser. What do you see in this patient? behavioral testing of clinical NLP models. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*, pages 63–73, Seattle, WA, July 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.clinicalnlp-1.7.

B. van Aken, J. Papaioannou, M. G. Naik, G. Eleftheriadis, W. Nejdl, F. A. Gers, and A. Löser. This patient looks like that patient: Prototypical networks for interpretable diagnosis prediction from clinical text. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2022 - Volume 1: Long Papers, Online Only, November 20-23, 2022*, pages 172–184. Association for Computational Linguistics, 2022b.

L. van der Maaten. Learning a parametric embedding by preserving local structure. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009*, volume 5 of *JMLR Proceedings*, pages 384–391. JMLR.org, 2009.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

N. Verma, K. Murray, and K. Duh. Strategies for adapting multilingual pre-training for domain-specific machine translation. In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track), AMTA 2022, Orlando, USA, September 12-16, 2022*, pages 31–44. Association for Machine Translation in the Americas, 2022.

J. Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, pages 37–42. Association for Computational Linguistics, 2019.

F. D. Vigna, A. Cimino, F. Dell'Orletta, M. Petrocchi, and M. Tesconi. Hate me, hate me not: Hate speech detection on facebook. In *Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), Venice, Italy, January 17-20, 2017*, volume 1816 of *CEUR Workshop Proceedings*, pages 86–95. CEUR-WS.org, 2017.

V. Vincze, G. Szarvas, R. Farkas, G. Móra, and J. Csirik. The bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinform.*, 9(S-11), 2008.

E. M. Voorhees. Overview of TREC 2001. In *Proceedings of The Tenth Text REtrieval Conference, TREC 2001, Gaithersburg, Maryland, USA, November 13-16, 2001*, volume 500-250 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2001.

E. Wallace, Y. Wang, S. Li, S. Singh, and M. Gardner. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5306–5314. Association for Computational Linguistics, 2019.

Y. Wan, B. Yang, D. F. Wong, Y. Zhou, L. S. Chao, H. Zhang, and B. Chen. Self-paced learning for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1074–1080. Association for Computational Linguistics, 2020.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

W. L. Warner and J. Hirschberg. Detecting hate speech on the world wide web. In *LSM@ACL*, 2012.

Z. Waseem. Are you a racist or am I seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science, NLP+CSS@EMNLP 2016, Austin, TX, USA, November 5, 2016*, pages 138–142. Association for Computational Linguistics, 2016.

Z. Waseem and D. Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the Student Research Workshop, SRW@HLT-NAACL 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 88–93. The Association for Computational Linguistics, 2016.

Z. Waseem, J. Thorne, and J. Bingel. Bridging the gaps: Multi task learning for domain transfer of hate speech detection. *Online Harassment*, pages 29–55, 2018.

J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

R. Weischedel, E. Hovy, M. Marcus, M. Palmer, R. Belvin, S. Pradhan, L. Ramshaw, and N. Xue. *OntoNotes: A Large Training Corpus for Enhanced Processing*. Springer, Heidelberg, 2011.

W. Wen, Y. Liu, C. Ouyang, Q. Lin, and T. L. Chung. Enhanced prototypical network for few-shot relation extraction. *Inf. Process. Manag.*, 58(4):102596, 2021.

P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proc. IEEE*, 78(10):1550–1560, 1990.

J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

L. Wittgenstein and G. E. M. Anscombe. Philosophische untersuchungen = philosophical investigations. 1953.

T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics, 2020.

S. Wu, T. Miller, J. Masanz, M. Coarr, S. Halgrim, D. Carrell, and C. Clark. Negation's not solved: generalizability versus optimizability in clinical natural language processing. *PLoS One*, 11(9), 2014.

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young,

J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

E. Wulczyn, N. Thain, and L. Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1391–1399. ACM, 2017.

G. Xiang, B. Fan, L. Wang, J. I. Hong, and C. P. Rosé. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 1980–1984. ACM, 2012.

P. Xie, H. Shi, M. Zhang, and E. P. Xing. A neural architecture for automated ICD coding. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1066–1076. Association for Computational Linguistics, 2018.

Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy. Hierarchical attention networks for document classification. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1480–1489. The Association for Computational Linguistics, 2016.

Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics, 2018.

Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764, 2019.

D. Yin and B. D. Davison. Detection of harassment on web 2.0. In *CAW2.0@WWW*, 2009.

M. Yuan, P. Xia, C. May, B. V. Durme, and J. L. Boyd-Graber. Adapting coreference resolution models through active learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7533–7549. Association for Computational Linguistics, 2022.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer, 2014.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, 2021a.

D. Zhang, J. Thadajarassiri, C. Sen, and E. A. Rundensteiner. Time-aware transformer-based network for clinical notes series prediction. In *Proceedings of the Machine Learning for Healthcare Conference, MLHC 2020, 7-8 August 2020, Virtual Event, Durham, NC, USA*, volume 126 of *Proceedings of Machine Learning Research*, pages 566–588. PMLR, 2020a.

H. Zhang, A. X. Lu, M. Abdalla, M. B. A. McDermott, and M. Ghassemi. Hurtful words: quantifying biases in clinical contextual word embeddings. In *ACM CHIL '20: ACM Conference on Health, Inference, and Learning, Toronto, Ontario, Canada, April 2-4, 2020 [delayed]*, pages 110–120. ACM, 2020b.

J. Zhang, J. Zhu, Y. Yang, W. Shi, C. Zhang, and H. Wang. Knowledge-enhanced domain adaptation in few-shot relation classification. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 2183–2191. ACM, 2021b.

Q. Zhang and S. Zhu. Visual interpretability for deep learning: a survey. *Frontiers Inf. Technol. Electron. Eng.*, 19(1):27–39, 2018.

S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. T. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068, 2022.

Z. Zhang and L. Luo. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semantic Web*, 10(5):925–945, 2019.

Z. Zhang, D. Robinson, and J. A. Tepper. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 745–760. Springer, 2018.

Z. Zhang, J. Liu, and N. Razavian. BERT-XML: large scale automated ICD coding using BERT pretraining. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop, ClinicalNLP@EMNLP 2020, Online, November 19, 2020*, pages 24–34. Association for Computational Linguistics, 2020c.

Y. Zhao, Q. Hong, X. Zhang, Y. Deng, Y. Wang, and L. R. Petzold. Bertsurv: Bert-based survival models for predicting outcomes of trauma patients. *CoRR*, abs/2103.10928, 2021a.

Z. Zhao, Z. Zhang, and F. Hopfgartner. A comparative study of using pre-trained language models for toxic comment classification. In *Companion of The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 500–507. ACM / IW3C2, 2021b.

H. Zhong, H. Li, A. C. Squicciarini, S. M. Rajtmajer, C. Griffin, D. J. Miller, and C. Caragea. Content-driven detection of cyberbullying on the instagram social network. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3952–3958. IJCAI/AAAI Press, 2016.

Q. Zhu, X. Chen, P. Wu, J. Liu, and D. Zhao. Combining curriculum learning and knowledge distillation for dialogue generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 1284–1295. Association for Computational Linguistics, 2021.

S. Zimmerman, U. Kruschwitz, and C. Fox. Improving hate speech detection with deep learning ensembles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA), 2018.

G. K. Zipf. Human behavior and the principle of least effort. 1949.