# Analysis and Domain-Transfer of Transformer Language Models in Limited Data Environments

Benjamin Winter

Dissertation
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)
der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel
eingereicht im Jahr 2024

**Erklärung entsprechend §9 der Promotionsordnung**

Hiermit gebe ich folgende Erklärungen ab:

- ▻ Diese Abhandlung ist, abgesehen von der Beratung durch die Betreuerin oder den Betreuer, nach Inhalt und Form meine eigene Arbeit.
- ▻ Ich habe keinen Teil dieser Arbeit bereits an anderer Stelle im Rahmen eines Prüfungsverfahrens vorgelegt. Teile wurden, wie in der Arbeit gekennzeichnet, im Rahmen wissenschaftlicher Veröffentlichungen publiziert.
- ▻ Die Arbeit ist unter Einhaltung der Regeln guter wissenschaftlicher Praxis der Deutschen Forschungsgemeinschaft entstanden.
- ▻ Es wurde mir noch nie ein akademischer Grad entzogen.

Kiel,

# Zusammenfassung

Diese Dissertation diskutiert die internen Prozesse von Transformer Sprachmodellen sowie Möglichkeiten für ihren effizienten Domänentransfer. Im Zentrum dieser Forschung steht das Bestreben, Verständnis darüber zu erlangen, wie diese Modelle Sprache verarbeiten und repräsentieren. Zu diesem Zweck führen wir qualitative und quantitative Analysen durch und vergleichen die internen Prozesse dieser Modelle mit der traditionellen NLP-Pipeline. Anschließend wenden wir dieses Verständnis an, um zwei unterschiedliche Ansätze des Transferlernens zu untersuchen, die speziell für Nischendomänen mit begrenzter Datenverfügbarkeit zugeschnitten sind. Insbesondere evaluieren wir unsere Methoden anhand der klinischen Domäne.

Der erste Ansatz stellt eine neuartige Trainings-Methode basierend auf Generierung von Wissensgraphen dar, und hat das Ziel die Lücke zwischen allgemeinem Sprachverständnis und domänenspezifischen Fakten sowie idiosynkratischer Sprache zu überbrücken. Durch die Anreicherung von Transformer-Modellen mit sorgfältig kuratiertem Wissen überwindet dieser Ansatz die inhärenten Herausforderungen, die durch den Mangel an spezialisierten Trainingsdaten entstehen.

Der zweite Ansatz setzt auf die Nutzung von Reinforcement Learning als Mittel zum fine-tuning von Transformer-Modellen. Dieser Ansatz nutzt die interaktive und iterative Natur des Reinforcement Learning, um Modelle effektiver an die Besonderheiten von domänenspezifischen Anwendungen anzupassen, insbesondere in Szenarien, in denen Daten spärlich sind. Dadurch, dass Modelle darauf trainiert werden, Entscheidungen und Vorhersagen zu treffen, die durch domänenspezifische Ziele und Rewards informiert sind, ist dieser Ansatz prädestiniert für die Anwendung in kritischen Bereichen wie der Differentialdiagnose.

# Abstract

This thesis discusses the intricate internal workings of transformer language models, as well as avenues for their efficient domain transfer. At the heart of this research is an endeavor to enhance our understanding of how these complex models process and represent language. To that end we conduct qualitative and quantitative analyses on these models and compare them to the traditional NLP pipeline. We then apply this understanding to investigate two distinct avenues of transfer learning, specifically tailored to niche domains characterized by limited data availability.

The first avenue explores the potential of enriching transformer models with domain-specific knowledge through the integration of information from knowledge graphs. This approach involves a novel method of targeted knowledge graph generation, aiming to bridge the gap between general language understanding, and domain-specific factoids and idiosyncratic language. By infusing transformer models with carefully curated knowledge, this strategy seeks to overcome the inherent challenges posed by the scarcity of specialized training data.

The second avenue shifts the focus towards leveraging reinforcement learning as a means to fine-tune transformer models for tasks within niche domains. This approach capitalizes on the interactive and iterative nature of reinforcement learning to adapt models more effectively to the peculiarities of domain-specific applications, even in scenarios where data is sparse. By training models to make decisions and predictions that are informed by domain-specific objectives and rewards, this method shows promise in significantly enhancing model performance in critical areas such as differential diagnosis and treatment recommendation.

Throughout this thesis, the efficacy of these transfer learning strategies is rigorously evaluated in the clinical domain, and their potential to substantially improve the adaptability and accuracy of transformer models in niche domains is demonstrated.

# Preface by the Author

## Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisors, Prof. Alexander Löser and Prof. Ralf Krestel, for their guidance throughout the course of my doctoral research. Their invaluable insights and expertise have been instrumental in shaping this work, and I am deeply grateful for their mentorship.

I would also like to extend my sincere thanks to Professor Felix Gers and Professor Amy Siu for their thoughtful feedback and contributions. Their perspectives have greatly enriched my research and have been crucial in advancing my understanding of the field.

My heartfelt appreciation further goes out to my fellow PhD students, Betty van Aken, Alexei Figueroa, Tom Oberhauser, Paul Grundmann, Jens-Michalis Papaioannou and many others who have shared this journey with me. Their camaraderie and collaboration have made this challenging endeavor a rewarding experience.

To my family—Petra Winter, Alexander Winter-Musiol, and Julia Winter—your patience, encouragement and support have been my foundation throughout this journey. Your belief in me has been a constant source of strength and motivation.

Last but not least, I would like to thank my girlfriend, Maria Miura, for her endless love, understanding, and encouragement. Her steadfast support has been my anchor during the most challenging times of this journey.

To all of you, thank you for making this thesis possible. Your contributions, both big and small, have been invaluable, and I am eternally grateful.

## Publikationen

Diese Dissertation basiert auf einer Reihe wissenschaftlicher Artikel, die auf verschiedenen Konferenzen veröffentlicht wurden. Diese werden im Folgenden aufgelistet, und anschließend beschreiben wir den Anteil der Arbeit, den die Hauptautoren und insbesondere Benjamin Winter an der Erstellung dieser wissenschaftlichen Artikel hatte. Die für die Analyse von Transformer-Sprachmodellen in Kapitel 3 relevanten Publikationen sind insbesondere::

- Betty van Aken[1], **Benjamin Winter**[1], Felix A. Gers and Alexander Loser. "How Does BERT Answer Questions? A Layer-Wise Analysis of Transformer Representations." In: ACM International Conference on Information and Knowledge Management (CIKM), November 2019. (Full Paper) [AWL+19]
- Betty van Aken[1], **Benjamin Winter**[1], Felix A. Gers and Alexander Loser. "VisBERT: Hidden-State Visualizations for Transformers". The Web Conference (WWW), 2020. (Demonstration Paper) [AWL+20]

und die für das Transferlernen und alternative Trainingsansätze für neuronale Netze relevanten Publikationen, die in den Kapiteln 4 und 5 diskutiert werden sind:

- **Benjamin Winter**[1], Alexei Figueroa Rosero[1], Alexander Löser, Felix Alexander Gers, and Amy Siu. “KIMERA: Injecting Domain Knowledge Into Vacant Transformer Heads”. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference, June 2022 (Full Paper) [WRL+22]
- **Benjamin Winter**[1], A. Figueroa[1], A. Löser, F. A. Gers, and Ralf Krestel. “DDxGym: Online Transformer Policies In a Knowledge-Graph Based Natural Language Environment”. In: LREC-COLING May, 2024. (Full Paper) [WFL+23]

[1]Die zwei ersten Autoren dieser Publikationen haben gleichwertige Arbeit zu dieser Publikation beigetragen.

# Contents

**Preface by the Author** ix

## I Preliminaries 3

**1 Introduction** 5
1.1 Motivation . . . 5
1.1.1 Efficient Adaptation of Text Embeddings . . . 6
1.1.2 Challenges in Domain Specific Language Transfer . . 9
1.2 Research Questions . . . 11
1.3 Contributions . . . 14
1.4 Thesis outline . . . 17

**2 Foundation** 19
2.1 Transformer Language Models . . . 19
2.2 Domain Adaptation . . . 21
2.3 Advances in Transformer Architecture . . . 25
2.4 Large Language Models . . . 26

## II Interpretability and Domain-Adaptation of Transformer Language Models 33

**3 Analyzing the Internal Processes of Transformers** 35
3.1 Introduction . . . 35
3.2 Related work . . . 37
3.3 Methodology . . . 38
3.3.1 Analysis of Transformed Tokens . . . 38
3.3.2 Probing BERT's Layers . . . 39
3.4 Datasets and Models . . . 43
3.4.1 Datasets . . . 43

3.4.2 Models . . . 45
3.4.3 Applying BERT to Question Answering . . . 45
3.5 Experiments and Results . . . 46
3.5.1 Phases of BERT's Transformations . . . 48
3.5.2 Comparison to GPT-2 . . . 56
3.5.3 Additional Findings . . . 56
3.6 VisBERT . . . 60
3.7 Limitations . . . 61
3.8 Summary . . . 61

**4 Efficiently Integrating Structured Knowledge Into Generic Transformer Models 65**
4.1 Introduction . . . 65
4.2 Related Work . . . 67
4.3 Methodology . . . 69
4.4 Datasets and Downstream Tasks . . . 74
4.4.1 Knowledge Graphs . . . 75
4.4.2 Clinical Answer Passage Retrieval(CAPR) . . . 76
4.4.3 Clinical Outcome Prediction(COP) . . . 76
4.5 Experiments and Results . . . 77
4.5.1 Models and Baselines . . . 78
4.5.2 Clinical Answer Passage Retrieval . . . 79
4.5.3 Clinical Outcome Prediction . . . 81
4.5.4 General Language Understanding (GLUE) . . . 82
4.5.5 Additional Experiment: Common-Sense . . . 83
4.6 Discussion and Analysis . . . 86
4.7 Limitations . . . 89
4.8 Summary . . . 89

**5 A RL Environment for Differential Diagnosis and a Novel Learning Strategy to Solve It 91**
5.1 Introduction . . . 91
5.2 Related Work . . . 94
5.2.1 RL in Automated Diagnosis Systems . . . 94
5.2.2 Structured Medical Knowledge . . . 95
5.2.3 Reinforcement Learning using Transformers . . . 95

5.3 DDxGym Environment . . . . . 96
5.3.1 Environment Definition . . . . . 97
5.3.2 DDxGym-Knowledge Graph . . . . . 100
5.4 Methodology . . . . . 102
5.4.1 Algorithm . . . . . 103
5.4.2 Transformer Encoder . . . . . 104
5.4.3 Additional Learning Objectives . . . . . 104
5.5 Experimental Setup . . . . . 106
5.6 Experiments and Results . . . . . 110
5.6.1 Initial Experiment: Project Hospital Data . . . . . 110
5.6.2 Quantitative Results on DDxGym . . . . . 112
5.6.3 Additional Experiment: Action Embeddings . . . . . 113
5.6.4 Additional Experiment: Fruitfly . . . . . 116
5.7 Discussion and Analysis . . . . . 119
5.8 DDxGym Demonstrator . . . . . 122
5.9 Limitations . . . . . 123
5.10 Summary . . . . . 125

**III Closing Discussion** **127**

**6 Review of Conducted Research** **129**
6.1 Review of the Research Questions . . . . . 129
6.2 Limitations of Presented Work . . . . . 133

**7 Outlook** **137**
7.1 Business Perspectives . . . . . 137
7.2 Future Work . . . . . 139

**8 Conclusion** **143**

**A List of Utilized Software** **151**
A.1 Programming . . . . . 151
A.2 Writing . . . . . 151

**Bibliography** **161**

# Part I

# Preliminaries

Chapter 1

# Introduction

## 1.1 Motivation

Large Language Models(LLMs) have drastically transformed the NLP landscape in the last few years. Dialog interfaces such as ChatGPT have given direct access to such models to a broad audience of both experts and more importantly people that did not previously engage with NLP models directly. Anyone can now use them to aid in a vast number of different tasks, such as summarization, paraphrasing, generation of creative texts, code, and many others. And, while far from perfect[1], they exhibit impressive performance. Because of both the ease of access, and this performance, they have nearly entirely usurped other architectures for many applications. Yet, we know very little about how these models actually function, and their adoption in critical domains such as medicine is almost non-existent. In this thesis we will discuss what we think are the main reasons for this slow adoption, and provide pathways and approaches that make them more suitable for such applications.

The basis for these models is the Transformer[VSP+17] architecture. It achieved initial success only in machine translation. In this task it handily beat previous architectures such as LSTM models. A big jump in popularity of this architecture however was made possible by BERT[DCL+19]. Through a bidirectional architecture, and a novel pre-training approach the authors created a model that could be trained once using vast data resources and computation, and then subsequently be fine-tuned efficiently on a wide variety of generic classification and regression tasks using only little data and computation. This has been the dominant approach to solv-

[1] https://www.reuters.com/legal/new-york-lawyers-sanctioned-using-fake-chatgpt-cases-legal-brief-2023-06-22/

ing NLP since then. And while some key advances have been made, such as reducing the time complexity of transformations[WLK+20], increasing the very limited input sequence length[BPC20], and improving the transfer learning efficiency with zero-shot and similar approaches[CHH+23], the architecture and process that are most commonly used largely haven't changed. It can even be argued, that the majority of the performance improvements to these models in the last few years, are owed to computation alone, not actual feature innovations[HBE+24].

While the process of adapting Transformer Language Models for specific applications as previously described might seem straight-forward, it can present unique challenges. This is particularly true when aiming to adapt a model to a niche domain that is very different from the generic training data the model has encountered during pre-training, and the problem is exacerbated in domains with very limited training data to begin with. In the conventional supervised learning paradigm data scarcity often considerably hinders model performance. This dissertation will therefore put a large focus on *efficient* models and approaches. The following sections will highlight what building such efficient models entails and what these challenges are in detail, and with that further motivate this dissertation.

### 1.1.1 Efficient Adaptation of Text Embeddings

In NLP research, there has been an ever-growing push towards larger models, trained on more hardware, and fed with more data, in the hope that a powerful enough general model easily adapts to any task and domain. While this has been shown to hold true to some degree, we argue that this class of model struggles in two key areas, which limit their application in many real-world scenarios. Throughout this thesis these areas are discussed in detail:

1. **Efficiency**. The aforementioned large models are becoming increasingly prohibitive in their proper usage and training to only the largest companies in the world, due to computational resource and data constraints. $GPT-4$ [OA+23] for example, one of the strongest language models at the time of writing and popularized through the chatbot interface Chat-

GPT Plus has cost OpenAI more than $100 million dollars to train[Kni]. While of course the fine-tuning and inference of such models is cheaper by a large margin, it is still intractable for many entities. A further problem is data efficiency. These large models require vast amounts of data to be trained optimally, and not for every application and domain it is tractable to collect such data in large enough quantities. And while these large models boast impressive amounts of parameters in the billions or possibly even trillions, not all of those parameters are even actively or optimally utilized[MLN19]. Efficient adaptation strategies should seek to harness these latent parameters optimally, repurposing them for domain-specific tasks without bloating the model further. This could lead to overall smaller models, with faster training times, and less associated costs.

2. **Explainability** stands out as another non-negotiable attribute, especially when adapting embeddings for critical applications e.g. in the clinical domain. Users and stakeholders need to understand, trust, and be able to critique the model's decisions in order for them to be applied in such domains. However, Deep Learning models are notoriously "black boxes", where the internal workings are difficult to interpret. Specific steps need to be taken to remedy that. One approach to gaining more understanding of these models is the analysis of their internal processes and transformations. A second approach is to have the model explain its own outputs with additional information. Both of these approaches will be discussed further in this thesis.

With these key areas in mind, we investigate two major avenues to improve on the simplistic paradigm of pre-training and fine-tuning that has become commonplace in NLP and which has brought about these generic transformer language models:

1. **Structured data**. One avenue to increase the efficiency of language models is the integration of structured data, particularly from knowledge graphs and knowledge bases. Such structured sources provide a rich network of relations and hierarchies, offering a robust foundation for text embeddings to capture nuanced relationships even when raw text

data is limited. Not only does this approach expose entirely new data sources not usually utilized with Large Language Models, this dense and relational data might be able to yield more efficient training since each data point contains more meaningful information when compared to the unlabeled text used in (Masked) Language Model Training. When integrating the structured data, one objective will also be to increase parameter efficiency by reusing parameters optimally.

2. **Learning Paradigms**. Another vision, one which has recently gained more traction, posits that models need more dynamic learning environments to truly capture the essence of data. Instead of the classic supervised learning paradigm where models simply map each data point to its label, introducing interaction and sequential decision-making can create a more holistic learning environment. Specifically, we investigate using RL as an alternative pre-training and fine-tuning strategy. In such an RL setting, models are encouraged to understand sequences, relationships, consequences, and the evolution of data points over time, promoting a deeper comprehension of the underlying domain. This vision emphasizes the importance of models "actively" learning from their environment rather than passively fitting to static datasets, which has been shown to lead even very successful and powerful models to overfit on statistical anomalies in the data, rather than abstracting the actual problem[NK19].

In summary, the efficient adaptation of text embeddings has to extend beyond optimizing the popular approach of Language Modelling Pre-Training and Downstream Task fine-tuning. We aim to explore novel training paradigms for these models, new tasks, and new forms of data. This dissertation will discuss in particular the issue of explainability of large language models, and will then address the challenges that come with transfer learning in low data, niche domains. In order to address these challenges we will discuss how to introduce structured data in the form of knowledge bases to these models, as well as the use of Reinforcement Learning with such models as two distinct avenues. But first, we will describe in greater detail the actual challenges these models face in these domains.

### 1.1.2 Challenges in Domain Specific Language Transfer

In the ever-evolving landscape of machine learning, language models have marked a significant milestone in our ability to process and generate human-like text. However, despite their abilities, most of these models are created to be powerful but generic, often trained on vast amounts of diverse textual data from books or the web. This generic nature occasionally falters when encountering idiosyncratic language, especially in specialized domains like medicine, which will be used for the remainder of this dissertation to illustrate the challenges and advances in domain adaptation. We identify 5 distinct challenges that come with domain adaptation in these domains.

1. **Lack of Data**. One of the most crucial challenges in applying domain-specific transfer learning, particularly in niche fields like medicine, is the constraint of limited data. The medical domain, characterized by its highly specialized and technical nature, often lacks the vast and diverse datasets that are readily available in more general domains. This scarcity is further compounded by privacy concerns and the sensitive nature of medical data, which restricts the volume of information that can be shared or accessed for research purposes. Such a limited dataset not only hampers the ability of models to learn nuanced and domain-specific patterns but also poses a particular challenge in training deep learning models like transformers, which typically require large amounts of data to achieve optimal performance. Further complicating this is the fact that different types of data or different datasets can be complementary but are not always so. Often, datasets stem from different sources, each with its own biases, scopes, and idiosyncratic language. This makes it challenging to combine data from e.g. different hospitals, into one bigger dataset. Moreover, data on Long-Tail diseases, though crucial, is by definition sparse, posing a significant challenge for models that thrive on large volumes of data. This lack of data is the predominant issue that underpins most of the other challenges outlined here, and the one that will be most eminently addressed in this dissertation.

2. **Idiosyncratic Language**. Another primary hurdle is that Medical language is not just a subset of everyday language; To encapsulate holistic

patient representations, and a large variety of different diseases, symptoms and procedures, a model requires specific domain terminology. While even generic models can be applied with some minor success to a wide range of tasks in the medical domain, ranging from symptom identification, triage, Differential Diagnosis (DDx), to pinpointing Medical Markers, the uniqueness and specificity of the domain's language introduce multifaceted challenges.

3. **Multi-modal, sparse, and complex data**. A secondary problem is the inherent complexity of medical data. Unlike typical text data, medical information is multi-modal, comprising not just flowing text, but also time-variated readings, structured lab results, and diagnostic images. The time-variated nature of some data in particular adds another layer of intricacy, demanding models to understand and process sequences of events, temporal relationships, and their implications.

4. **Decision Legitimization**. Additionally, for a medical professional, the stakes are incredibly high. Decisions and recommendations derived from models must not only be accurate but also understandable. While in some fields an accurate prediction might be enough, in medicine the "why" behind a decision can be as vital as the decision itself. This holds especially true, since the goal in this domain should not be to replace doctors, but to aid them in their decision-making. Traditional methods for model explainability often fall short in capturing the nuances of specific problems, and offer only vague guidance as to what a deep model is "thinking". It is therefore critical to understand how these models work internally, to increase the trust we can put in them.

5. **Efficiency**. Finally, **efficient** adaptation becomes paramount. In a domain where rapid and accurate decisions can mean the difference between life and death, models need to be efficient, both in terms of computational parameters and training times, in order to be able to incorporate new data as often as possible, as well as respond adequately in emergencies. This holds especially true since hospitals usually have very limited access to hardware, so scaling up models to increase their performance is simply not a viable option.

In summary, while the potential of language models in domain-specific

applications like medicine is immense, the road to their effective and reliable deployment requires additional considerations. Specifically we need to search for complementary data and context that ideally explores new data sources beyond plain text, and we need a particular focus on explainability, both of which will be addressed in the further chapters of this thesis.

## 1.2 Research Questions

The research questions discussed in this thesis follow the overarching question of "How can we efficiently adapt generic transformer language models to real-world applications of niche domains?". Because there are many challenges in these domains as outlined in the previous section, this question requires a multi-faceted answer. In particular, the research presented here focuses on the facets of Interpretability and Explainability, as well as efficient Transfer Learning for small data environments.

In order to adapt and improve on generic transformer language models, we need to first understand them. To that end the first research question addressed by chapter 3 aims to provide a new avenue of interpreting the internal workings of Large Transformer Language Models. Then, questions two and three are addressed by chapters 4 and 5 respectively. They explore two distinct avenues of efficient domain adaptation, guided by the understanding gained from the study of the first question, as well as similar research. These two avenues are chosen uniquely for their applicability to niche domains like the clinical domain, which are usually characterized by a severe lack of training data. Specifically, research question 2 will address the data limitation problem by making use of structured data as an additional data resource. Research question 3 on the other hand will address that problem by generating an infinite stream of data via a RL environment.

## Research Question 1: Do Transformer Models reconstruct the NLP Pipeline in their Layers?

As discussed before, transformer models, like any deep learning model, are largely black boxes. In order to understand them, we need to take a look at the transformations happening at each layer of their deep structure. Additionally, they present a large break from more traditional NLP approaches. Previously, in order to solve a complex task, multiple different models and algorithms had to be chained together and feature engineering had to be conducted. These traditional stages range from part-of-speech tagging, semantic and syntactic parsing, and named entity recognition, and lead to downstream tasks such as question answering and relation extraction. Now, just a single powerful transformer model can solve these complex downstream tasks handily in just one step. Naturally this question then arises: Do these models actually perform these traditional steps implicitly and internally, or do the transformations follow entirely different and more abstract processes? Answering this question holds significant implications. Not only does it demystify the internal mechanisms of state-of-the-art models and thereby increase the trust we can place in them, but it might also point us in the direction of how to best improve these models, where they fail, and how to specifically target their adaptation.

## Research Question 2: Can over-parameterised models be improved through targeted Knowledge Graph Completion Retraining?

In the field of machine learning, the design philosophy behind deep networks, especially transformers, has trended towards vast over parameterization, with models often consisting of millions or now even billions of parameters. Because of this enormous amount of parameters, there's an emerging consensus that a significant portion of these parameters remains largely dormant or underutilized throughout typical training regimes and for typical downstream tasks. As discussed previously, this stands in the way of both efficient adaptation, and explainability. In order to apply these powerful models to niche domains, with limited data and hardware, we need to reduce over parameterization and break the per-

formance to parameter count dependency. Further, we need to instill the distinct, domain-specific knowledge into these models, that they could not gain through the generic language modelling pre-training. Additionally, we need to be able to train models with more data than is available as unstructured text in niche domains. One compelling approach which potentially addresses all of these issues is the application of knowledge graph completion as a method of retraining. Domain-specific Knowledge Graphs present an underutilized resource as training data for Transformer language models. Their structure also makes their information richer and more specific than plain text. Transformers have also been shown already to be a suitable architecture for knowledge graph completion [BRS+19]. In this dissertation we will therefore explore methods to utilize Knowledge graphs to instill domain-specific knowledge in these models. Using knowledge we gained through research of the first research question we will strategically target specific parts of the model for this training to combat over parameterization and prevent the loss of general language capabilities known as catastrophic forgetting.

### Research Question 3: Is RL a suitable alternative to supervised learning in the Differential Diagnosis scenario?

In the current machine learning landscape, supervised and semi-supervised learning dominate, particularly due to popular classification tasks and benchmarks, as well as (Masked) Language Modelling being a very prominent and powerful training tool. Their direct approach of mapping inputs to outputs often leads to impressive and consistent results across varied domains. However, research has shown that this learning from simplistic mapping leads the models to questionable abstractions. Instead of learning concepts, relations and thought processes that would enable them to solve a problem like a human would, they instead rely on dubious statistical anomalies, and exploit them to the fullest[NK19]. This is deeply integral to how these models learn and leads to them being easily fooled in adversarial attacks. Therefore, in an effort to train more sensible models which make decisions more akin to humans, we explore the alternative training paradigm of RL. However, the main reason we choose to explore RL is that it elegantly solves the limited data problem, as we will be able to

generate new samples through the RL environment with only little initial labelling required. RL emphasizes sequential decision-making, introduces time horizons and consequences for the model's training. This makes it a promising approach to solve a complex task like Differential Diagnosis. In the task of DDx it is not sufficient to simply classify a diagnosis from an input, since with limited information at the admittance of a patient that is scarcely possible. Instead, a sequential back and forth is required where the patient is examined, and different procedures are applied. For these reasons we choose DDx to measure our efforts in this avenue of research. And eventually, the trajectories followed by an ideal agent could even lead to novel medical insights on how to most effectively treat certain conditions.

## 1.3 Contributions

This chapter presents an overview of our efforts and contributions, which align closely with the research objectives previously outlined. Thus, these contributions will address the question "How can we make transformer models more applicable in real world applications of niche domains?". To that end the first cornerstone of contributions is the in-depth analysis of transformer language models. We had previously identified the lack of interpretability and accountability as one of the major roadblocks in adoption of such models. A thorough analysis of the internal processes seeks to remedy that. This analysis is followed by the second cornerstone of this thesis, improving transformer models' adaptability to specialized domains. In order to adequately address the question that means approaches that are at least data efficient, but ideally both data and computation efficient. Concretely, we have explored the clinical domain, a realm characterized by its complexity and nuanced linguistic attributes, as a primary case study to showcase the potential of domain adaptation. The breakthroughs and methodologies documented here, while grounded in healthcare, are anticipated to be transferable and similarly impactful across a spectrum of other fields.

## A Layer-Wise Analysis of Transformer Representations

The first set of contributions of this thesis addresses research question 1 and thereby the analysis of the processes that happen in deep transformer language models.

- ▻ With a focus on the complex question answering task as a downstream task a suite of probing tasks is developed and applied to explore inner workings of transformer models. These probing tasks highlight different abilities of the tested models, and mirror steps in the NLP pipeline.
- ▻ We perform a qualitative analysis on the hidden representations of BERT via dimension reduction techniques such as Principal Component Analysis.
- ▻ We perform both qualitative and quantitative analysis 3 very different question answering datasets. One generated toy dataset in bAbI with very simple vocabulary, One real-world dataset from Wikipedia with SQuAD, and one especially challenging real-world dataset requiring multi-hop reasoning with HotpotQA.
- ▻ We demonstrate through both the qualitative analysis as well as the probing tasks different phases and processes at different layers of the network
- ▻ A visual demonstrator of our qualitative analysis is made available online for interactive exploration at `https://visbert.demo.datexis.com`

## Injecting Structured Domain Knowledge into Underutilized Attention Heads

The second set of contributions addresses research question 2. In particular, they detail our efforts on the problem of domain-adaptation, and the question of how to integrate structured knowledge from knowledge graphs in generic transformer language models. Incorporating structured knowledge graphs as intermediate transfer learning data opens up new data sources that many approaches currently do not use. This can greatly increase the data efficiency.

- ▻ We develop a novel knowledge graph completion task, which extends previous work on KG completion, and which had to the best of our knowledge never been used before as an intermediary re-training task.
- ▻ We utilize a model compression algorithm as a way to evaluate the importance of individual attention heads to a specific downstream task, and use this information in a novel way, to inform the targeted retraining of transformer language models.
- ▻ We develop multiple strategies to make use of this head importance information, among them a hard filtering of the heads, and a soft learning rate amelioration.
- ▻ We demonstrate the efficacy of these strategies as an approach to domain adaptation to the clinical domain in two different scenarios: (Multi-label-) Classification, and Zero-Shot Document Retrieval.
- ▻ We perform a qualitative analysis of how the attention head importance values change before and after the targeted retraining, and demonstrate that our approach significantly improves the importance of previous underutilized attention heads.

## Online Transformer Policies in a Knowledge Graph Based Natural Language Environment

The final set of contributions addresses research question 3, and concerns our work on a novel reinforcement learning environment for a common and complex clinical scenario, as well as our approach to make transformer language models significantly more viable as a standalone policy in an online Reinforcement Learning setting. Reinforcement Learning increases data efficiency even further, as very little initial data is necessary to generate a vast number of different trajectories the model can be trained on, as our research will show.

- ▻ We develop a novel reinforcement learning environment to model the differential diagnosis scenario, as well as the patient treatment process. This environment follows the OpenAI Gym guidelines and is thus widely and generically usable and applicable.

- ▻ As ground truth data for this environment we collate and label a knowledge graph spanning 111 diseases, as well as their symptoms, and the procedures necessary to reveal and treat these symptoms and diseases. We further label meta information for these diseases and symptoms like their severity and onset, in order to model Differential Diagnosis more realistically.
- ▻ We develop a novel training strategy for transformer networks as reinforcement learning policies, adding a parallel masked language modelling objective.
- ▻ We demonstrate that this objective stabilizes the otherwise very unstable transformer learning in online Reinforcement Learning. Our agent trained with this auxiliary objective easily solves more than half the diseases in our environment.
- ▻ We analyze why some diseases are harder to diagnose and treat than others, identifying examination overlap as the key factor that increases the problem complexity.
- ▻ We evaluate our created environment with a medical professional and demonstrate that the doctor follows similar trajectories as our best RL agent, while acting more strategic in long episodes and for the diseases where our agent fails.

## 1.4 Thesis outline

The remainder of this thesis is structured in the following way: Chapters 3, Chapter 4 and Chapter 5 contain each a published paper that arose from the research of this thesis and which comprise the main body of research done in the process of creating this thesis.

Chapter 3 corresponds to [AWL+19], Chapter 4 to [WRL+22] and Chapter 5 to [WFL+23]. These chapters attempt to answer the research questions that were posed in the previous chapter, and detail the research contributions that have been made during the creation of this thesis.

The concluding segment of the thesis begins with a comprehensive review of the research questions in Chapter 6.1. This is succeeded by Chap-

ter 6.2, where the constraints and limitations of the conducted research are critically examined and discussed.

Following this reflective evaluation, Chapter 7.1 delves into the business perspectives and potential commercial implications of the models and paradigms discussed throughout the thesis, offering a glimpse into their practical and economic significance. Further, Chapter 7.2 encapsulates the authors' prospective vision for the research area, highlighting potential pathways and future avenues of exploration and development in the field.

Finally, this thesis closes with Chapter 8, drawing conclusions and summarizing the pivotal findings of this thesis.

Chapter 2

# Foundation

In this section we outline the related work that underpins all of the research that is undertaken in this thesis. Each chapter will further, then go into detail about related research that is more immediately relevant, referencing this section where applicable. Specifically, we cover here the Transformer[VSP+17] architecture that builds the basis for all the models analyzed and developed in this thesis, as well as Domain Adaptation. Additionally, we highlight improvements on the transformer architecture and approaches to building efficient models, as they relate to our research. We further discuss major recent advances in Large Language Models, and advances in the efficient training of these, which run tangentially to our research.

## 2.1 Transformer Language Models

For the small transformer models that are the main subject of this thesis, there is a great number of different pre-trained models and architectures to choose from. These can be broadly categorized in two categories: Autoregressive Models i.e. Models using a Decoder, and Encoder-Only models.

### Autoregressive Models

Encoder-Decoder and Decoder-Only models like the GPT family of models [RN18; RWC+19; BMR+20; OAA+24] are trained in an autoregressive manner using language modelling. This makes them adept at text generation, which lead to their current popularity through interfaces such as ChatGPT. As the newest model at the time of our research we include

GPT-2 [RWC+19] in the analysis in the coming chapter. It represents OpenAI's improved version of GPT [Rad18] and while GPT-2 has not climbed leaderboards like BERT has, its larger versions have proven adept at the language modelling task. The larger examples of this family of models, e.g. [BMR+20; OAA+24; TLI+23; RBC+21; SFA+22] are unfortunately out of the scope of this thesis due to hardware and data limitations, which will further be discussed in section 6.2.

We give these models only little consideration in our research, and the majority of our work focuses on Encoder-only models. However, since all transformers share similarity in their building blocks, we expect our research to transfer to this category of models with little challenges.

### Encoder-Only Models

Encoder-Only transformer models, as the name suggests, lack a decoder and therefore can not be trained in an autoregressive manner. Instead, they make use of other pre-training tasks such as Masked Language Modelling and Next Sentence Prediction. The advantage of this category of models is that they can more easily be applied to a wide array of different classification and regression tasks. To that end one can simply remove the pre-training output layer(s) and add new output layers corresponding to the downstream tasks to be solved. This works significantly better than for autoregressive models due to both the bidirectional attention most Encoder-Only models implement, and the specific pre-training tasks chosen for these models.

Because of this architectural advantage in transfer learning we choose this category of models as the main object of our research. More concretely, we focus on the BERT family of models.

BERT[DCL+19] popularized this category of models and still represents one of the most used transformer models today[1]. Its popularity has spawned a great number of derivative models specialized in different languages and domains. These models share the exact same architecture, and only differ in the data used for pre-training, and, potentially, the tokenizer.

[1] https://huggingface.co/models?sort=downloads

Still, the generic English versions and in particular `BERT-base-uncased` are the most widely used. `BERT-base-uncased` with its 12 Encoder blocks, is strong enough to perform adequately well on downstream tasks, while being much easier and faster to fine-tune than its larger cousin `BERT-large-uncased`, which consists of more than three times as many parameters[DCL+19].

The aforementioned factors make `BERT-base-uncased` the obvious baseline for our research, and we use it as such, both for our analysis of transformers, and to evaluate our novel training approaches.

## 2.2 Domain Adaptation

*Domain Adaptation* represents one specialized area of Transfer Learning, in which either a model is trained on one domain, and then trained to apply to a second domain, or a model is trained on generic data spanning many domains, and then further specialized to one specific domain. We focus on the second case. Transfer Learning is especially popular with Transformer networks, due to large data and hardware requirements for pre-training, and the vast amount of pre-trained models available for use[PY09].

Popular examples of Domain Adaptation for transformer models include [XLS+19b; GMS+20], who continue pre-training models using Masked Language Modelling and Next Sentence Prediction on in-domain data, and [DSW+20] who add an adversarial domain distinguishing task.

All of these approaches rely heavily on unstructured text to solve the data scarcity issue, making use of the fact that unsupervised text data is usually abundant even for niche domains. The transfer learning approaches developed in this thesis however focus specifically on structured knowledge. The first of our approaches learns via knowledge graph completion, the second via Reinforcement Learning on an environment created on top of a knowledge graph.

There are three more particular sets of approaches to domain adaptation we want to highlight here: Zero-/Few-shot learning, Adapter Models, and Model Merging. While these fields of research only indirectly relate to our research, they follow the main goal of this thesis: building efficient domain specific models with little data and/or computation necessary. We

view them as the most promising alternatives to our work, in particular in the context of large language models and the current direction of that field of research.

## Zero-Shot and Few-Shot Learning

Few-shot and Zero-Shot learning methods address the challenge of data scarcity in the most concrete manner. For domain adaptation specifically that means that no, or only a handful of in-domain samples are seen by the model during training.

Few-shot learning methods employ techniques such as meta-learning[FAL17], which trains a model on a variety of learning tasks to promote quick adaptation to new tasks, and metric learning[BBB+93], which involves learning a distance function to compare and classify new data points effectively. These facilitate the application of predictive models to new, specialized domains with limited available data.

Zero-shot learning extends this concept further. It often leverages semantic relationships between known and unknown categories, and often uses auxiliary information such as class attributes or textual descriptions to bridge the gap between seen and unseen classes. [ALT+21] for example test GPT-3 and BART in a zero-shot setting based only on task instructions, and textual descriptions of the targeted labels. They show that even a powerful model such as GPT-3 struggles in this setting, performing worse than the plurality class baseline. Another example is [RT15] where a framework is developed which learns a mapping of input features to a set of attribute signatures for each class. At test time this mapping is used to classify new classes with unseen signatures to new classes.

While in the context of seeing few to no samples during training these approaches achieve impressive results, objectively the results are often mediocre, as different benchmarks show [XLS+19a; HZY+18; TZD+20]. Furthermore, these approaches naturally benefit from starting with a base model which already possesses the general (domain-)knowledge required to solve the few- or zero-shot tasks at hand. Therefore, approaches like the ones developed as part of this thesis can lead to better performance in these settings, by making use of alternative knowledge sources in the form of knowledge graphs to give the model a strong domain-specific prior.

Combining these ideas can then lead to a more holistic solution for the data scarcity problem.

### Adapter Models

The idea of Adapter Models is to extend a large generic model with one or multiple smaller adapter models, which enhance the original model's capabilities. They can for example provide support for new target languages in a translation model[BAC+22], or provide domain adaptation. The main advantage over regular fine-tuning is that only a small amount of parameters has to be trained, which eases both computational and data restraints, and enables easy multi-task training[HGJ+19; SM19]. It has further advantages for continual learning, as these adapters can be easily updated or swapped out should the need arise.

For domain adaptation specifically, [BF19] showcase a very simple and straightforward application of Adapters for Domain Adaptation. Their work still requires in-domain supervised data however, which might be hard to come by. This is solved by [MRK+23], who develop a method for unsupervised domain adaptation using adapters.

We expect future research in this area to focus on further optimizing the design of these adapters and addressing challenges related to their integration with complex base models. While such adapter models are clearly a powerful tool for efficient domain adaptation, they do still require adding further complexity and parameters to a potentially already large base model. In contrast, our approaches incorporate domain knowledge in models without modifying the architecture in any way. Our approaches thereby sacrifice the flexibility that adapter models provide, in favor of a more streamlined and straightforward architecture, and with that the option to apply our approaches to any transformer based architecture. With that in mind, our approaches could even be combined with the adapter architecture, though answering the question, whether they are complementary would require further testing.

## Model Merging

Model merging represents another recent approach in the field of transfer learning, with the specific aim of enhancing the efficiency and effectiveness of domain transfer. This technique involves combining two or more pre-trained models or their components to leverage their distinct strengths and bases of data, thereby creating a more robust model that can operate across different domains.

The concept of model merging extends from the broader framework of ensemble learning but focuses more on integrating learned features and behaviors from different models into a cohesive system, rather than keeping the different models in an ensemble separate. Early research in this area focused on simply averaging the weights of two or more neural network models [WIG+22; MR21]. From there, many unique approaches have been developed in a short span of time, the currently most popular being TRIM, ELECT SIGN & MERGE (TIES-Merging)[YTC+23] and DRop and REplace (DARE)[YYY+24]. TIES-Merging in particular is related to the research presented in chapter 4, since it too does domain transfer in a targeted way on only a specifically chosen subset of model parameters.

In the context of domain transfer and transfer learning, model merging offers a powerful way to combine a variable amount of models into one, spanning multiple domains, or even combining multiple models within one domain. It also largely follows the same paradigm of the models developed in this thesis by leaving the original model architecture unaltered. However, model merging has a steep requirement of trained, domain-specific models being freely available. For many niche domains that simply is not the case, rendering model merging entirely unusable in those domains. Especially in such cases, approaches like the ones detailed in this thesis will prevail.

To summarize, while model merging is a powerful tool for domain adaptation, and a relatively new field of research that is sure to see a myriad of innovations in the near future, our research has a different niche of applications.

### Medical and Clinical Transformer Models

As a specific instance of Domain Adaptation, Medical Transformer models are of special relevance to this thesis. There are a number of different Pre-Trained models for the medical domain already available, which should make the transfer learning for that domain substantially easier. The most popular one at time of research is BioBERT [LYK+20], which has been pre-trained directly on medical data. There are a number of similar models using different medical datasets, such as the work of [CBB+20].

These medical models will be used as additional baseline contenders in the coming chapters where applicable. Being pre-trained from scratch they however require both a large amount of in-domain data, and processing power in their creation, which is what this thesis aims to address. To that end we largely forego pre-training models ourselves in this thesis and instead provide solutions to making the most effective use of already pre-trained models.

## 2.3 Advances in Transformer Architecture

In the years since BERT's inception, many advancements have been made to its core architecture already. A large portion of these address some of the main shortcomings this architecture has, and thus run tangentially to the research presented in this thesis. We differentiate three categories of improvements.

- **Generalization** The first category improves the generalization capabilities. An example of this is the Universal Transformer[DGV+18] which seeks to add the recurrent inductive bias which made RNNs so powerful back to the linear transformer architecture.

- **Model Efficiency** The second category improves the efficiency of training and/or inference of these models. One example of this is the Linformer [WLK+20], which is a novel architecture that reduces the quadratic complexity of self-attention to linear space, making it more suitable for longer sequences without compromising performance. Other notable examples of this category, specifically spanning the topics Model Compression will more closely be discussed in chapter 4.

▻ **Sequence Length** The third category, unique to transformer models, attempts to lift the restrictions of sequence length that plague in particular Encoder-Only transformer models. Examples of this are TransformerXL [DYY+19] and the Longformer[BPC20], which manage to increase the sequence length from BERT's 512 tokens to effectively tens of thousands of tokens in the case of the Longformer.

While the aforementioned research achieved significant improvements over the basic BERT model, we stick with BERT as our baseline model. This is for two main reasons. Firstly, none of these other models in particular has risen in popularity enough to be a clear and undisputed upgrade. Closest to that is RoBERTa[LOG+19], which makes no changes to the model architecture, and achieves performance improvements through a change in pre-training hyperparameters. Secondly, changes in the base architecture as most of the previously outlined research exhibits, muddles the evaluation of our own contributions, making it more difficult to judge their actual benefit and complicating error analysis.

## 2.4 Large Language Models

While our research, focussing on small and efficient transformer models, stands separate from the body of research around LLMs, LLMs represent the current state-of-the-art of the transformer architecture. We therefore see it as productive to discuss recent developments in this subfield, and how they relate to the research presented here. It is difficult to determine where exactly to draw the line that differentiates transformers now seen as 'small', including BERT-base with 110 million trainable parameters, from models such as GPT-4, rumored to have around 1.7 trillion parameters[2], roughly 15000 times as much. For the purposes of this thesis however we understand Large Language Models as models with more than 1 billion parameters. We will briefly discuss major milestones in the development of these models.

GPT-3[BMR+20] scaled up the transformer-based model to a before unprecedented size (175 billion parameters). The authors demonstrated

[2]https://www.semafor.com/article/11/01/2023/microsoft-pushes-the-boundaries-of-small-ai-models

that a sufficiently big model can achieve significant performance in certain NLP tasks even with small amounts of task-specific data using few-shot learning. In the context of efficient models this is a double-edged sword. While good few-shot performance is something to strive for in low-data domains, the size of this model requires specific, dedicated hardware. Many even well supplied hospitals for example will not be able to serve such models on their infrastructure in the present or near future. A balance thus has to be struck between (data) efficient learning, and model size, which we aim to do with the research in this thesis.

GPT-4 [OAA+24] and its variants need no introduction. Scaling up GPT-3 by another factor of 10 lead to even more impressive Zero- and few-shot performance. These models now have very little to do with small transformer models in their capabilities and use. Generating very human-like text, and being able to answer a great variety of queries makes them very useful in a number of different scenarios. However, these models continue to struggle with reporting accurate facts, and can be overconfident in their answers. Also, due to their size and the restricted access, the only way to use them is to access OpenAI's servers. It is also not possible to fine-tune such models, unless working in a few select organizations that work together with OpenAI, and only in domains where extremely large datasets are available[3]. While the architecture is likely still largely similar to smaller transformer models, and thereby the methods developed in this thesis could theoretically be applied to such models, the use of that is questionable.

Lastly we want to highlight here BioMistral[LBM+24], based on the generic Mistral[JSM+23] language model. The Mistral training approach involves segmenting the model training process into smaller, more manageable parts, allowing for more efficient resource utilization. BioMistral specifically is a collection of Open Source Medical Large Language Models. Apart from the main 7 billion parameter model, multiple different variants are released with different quantization and merging strategies. While out of the scope of this thesis because of the size and its release only in the year 2024, it currently represents likely the most powerful openly available medical transformer model.

[3]https://openai.com/form/custom-models/

## Reinforcement Learning from Human Feedback (RLHF)

The NLP space is dominated by classification and generation problems, both of which tend to be challenging to frame as Markov Decision Processes. For many years chatbots and other dialogue systems were therefore the main applications for Reinforcement Learning in NLP. [OWJ+22a] however introduced a novel way to let NLP models profit from the benefits of RL, reinforcement learning from human feedback. With this approach a model is first trained in a supervised fashion. Then, model outputs are sampled and ranked by a human to their preference. These rankings are used to train a reward model, which is finally used to further train the model using RL.

The reinforcement learning applied in this way is very different from the one we apply in chapter 5, however. With RLHF there is no interaction with the environment, and the actions the model takes do not influence an environment, and subsequently the data it sees in its observations. An episode in RLHF represents a one-step process of generating an output from a sampled input, and getting a reward for it, much more akin to supervised learning. With the more canonical approach to RL that we follow, we build a model that is more explainable, since full decision trajectories can be followed. Additionally, we expect our model to learn more abstract knowledge about the environment and abilities beyond language understanding from the interactiveness of our environment.

RLHF does relate on the other hand to our approach in chapter 4. Both approaches make use of an intermediary re-training step, before the model is fine-tuned on the actual downstream task. In RLHF both the intermediary and fine-tuning tasks are the same from a human perspective though. In both cases the model generates text from a prompt. The difference lies in the scoring, first with a supervised loss and then with an RL loss from the reward model. In our approach both the data and the re-training task are different from the downstream task.

As a whole, RLHF addresses the data efficiency problem in a different way to the work we present here. Instead of integrating new types of data, or generating data in the form of RL trajectories, it approximates supervised labels via the reward model. This effectively turns unsupervised data into weakly or semi-supervised data.

### Retrieval Augmented Generation (RAG)

No matter the size of the transformer network used, one major concern remains the lack of explicit, factual data. On knowledge intensive tasks the implicit knowledge in their weights might fail, and updating such implicit knowledge when facts change or are added is still evolving research. Retrieval Augmented Generation (RAG)[LPP+20] aims to remedy those shortcomings. It enhances the strength of pre-trained language models by querying relevant information from an external corpus at runtime, and in particular with large language models it has seen a surge in popularity.

RAG involves adding additional orchestration and retrieval modules to an existing model, which comes at an efficiency cost, but as an alternative to full domain adaptation, it provides similar benefits to Adapter models. The difference between those again is that Adapters store knowledge implicitly, while in the RAG architecture the knowledge is stored explicitly. With Large Language Models the relative number of parameters that need to be added on top of the base model is vanishingly small, making it a very attractive approach.

In summary, RAG is a powerful alternative to the transfer learning approaches presented in this thesis. However it is different in spirit, favouring explicit over implicit knowledge, and favouring adding more parameters and a more complex architecture over a simple and streamlined model.

### Efficient Training of Large Language Models

Recent computational advances have significantly enhanced the efficiency of training transformer models, important in particular to Large Language Models, however smaller transformer models can benefit from them as well. One notable development is Low-Rank Adaptation (LoRA)[HSW+21], which increases training efficiency by adapting pre-trained models using low-rank matrices. That reduces the number of trainable parameters by magnitudes, while retaining respectable performance. It is demonstrated to be particularly effective in maintaining the integrity of pre-trained knowledge during fine-tuning, minimizing catastrophic forgetting.

In addition to LoRA, various quantization techniques have also played an important role in optimizing transformer models for efficient deploy-

ment. Quantization involves reducing the precision of the model's parameters, which can significantly decrease the computational cost and memory usage. Techniques such as quantization-aware training (QAT)[LOZ+23], which simulates low-precision arithmetic during training to minimize performance degradation, have been shown to reduce model size and speed up inference without substantial loss in accuracy.

Moreover, mixed-precision training, which uses both 16-bit and 32-bit floating-point arithmetic to balance computational efficiency with training stability, has been widely adopted for transformer models of all sizes. This method leverages the reduced memory footprint of 16-bit data types to accelerate computation while maintaining the numerical stability provided by 32-bit types. We make extensive use of this for our training.

These advancements collectively contribute to the broader effort of making transformer models more practicable for real-world applications by reducing their computational demands. As these technologies continue to evolve, they will undoubtedly unlock new possibilities for the deployment of advanced NLP models across various domains, especially in settings where computational resources are a limiting factor. They do not however address the data scarcity issues present in niche domains. These efforts run in parallel to the research presented in this thesis and combining such computational efficiency improvements with our research is logical.

# Part II

# Interpretability and Domain-Adaptation of Transformer Language Models

Chapter 3

# Analyzing the Internal Processes of Transformers

## 3.1 Introduction

As outlined previously, transformer language models, while powerful, still face challenges when it comes to their application in critical scenarios in the real world. One of them is their black box nature and thereby their lack of explainability and interpretability. Even with their impressive performance in a wide variety of tasks it is a difficult prospect to use such models in domains and applications that have a direct effect on human lives, when they can't be fully trusted or understood. The research presented here is meant to give a starting point to build that trust. In particular, we examine how these models work internally by offering a novel view into their hidden layers. We then compare those processes to the more established, traditional and well trusted NLP Pipeline approach. That approach enjoys better interpretability, because it makes use of multiple smaller and simpler models, which are then chained together. The secondary aim of this research is to uncover shortcomings of the transformer architecture, which further research in transfer learning can exploit.

While Transformers are commonly believed to be also somewhat interpretable through the inspection of their attention values, current research suggests that this may not always be the case [JW19]. We take a different approach. Instead of evaluating attention values, our approach examines the hidden states between encoder layers directly. We focus our efforts on the most popular architecture at the time of writing, the BERT [DCL+19] family of models, but give some indications of how this analysis might look like for other types of models.

3. Analyzing the Internal Processes of Transformers

Beyond research question 1 there are multiple questions this chapter will address:

1. Do Transformers answer questions decompositionally, in a similar manner to humans?

2. Do specific layers in a multi-layer Transformer network solve different tasks?

3. How does fine-tuning influence the network's inner state?

4. Can an evaluation of network layers help determine why and how a network failed to predict a correct answer?

We discuss these questions on the basis of fine-tuned models on standard QA datasets. We choose the task of Question Answering as an example of a complex downstream task that, as this chapter will show, requires solving a multitude of other Natural Language Processing tasks. Additionally, it has been shown that other NLP tasks can be successfully framed as QA tasks [MKX+18]. Therefore our analysis should translate to these tasks as well.

**Contributions**. We present the following contributions in this chapter:

First, we propose a layer-wise visualization of token representations that reveals information about the internal state of Transformer networks. This visualization can be used to expose wrong predictions even in earlier layers or to show which parts of the context the model considered as Supporting Facts.

Second, we apply a set of general NLP Probing Tasks and extend them by the QA-specific tasks of Question Type Classification and Supporting Fact Extraction. This way we can analyze the abilities within BERT's layers and how they are impacted by fine-tuning. These tasks are loosely modelled after the traditional NLP pipeline and will help us compare the two.

Third, we show that BERT's 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.

Fourth, we develop an openly available demonstrator with which our qualitative analysis can be reproduced.

## 3.2 Related work

The research in this chapter is heavily related to other research in Transformer Models and transfer learning as discussed at length in Section 2. We discuss here additionally relevant research in the topics of Interpretability and probing.

Explainability and Interpretability of neural models have become an increasingly large field of research. While there are a myriad of ways to approach these topics [Lip16; GMT+18; DBH18], we focus on relevant work in the area of research that applies probing tasks and methodologies, post-hoc, to trained models.

There have been a number of recent advances on this topic. While the majority of the current works aim to create or apply more general purpose probing tasks [CK18; BDD+17; SPK16], BERT specifically has also been probed in previous papers. [TXC+19] proposes a novel "edge-probing" framework consisting of nine different probing tasks and applies it to the contextualized word embeddings of ELMo, BERT and GPT-1. Both semantic and syntactic information is probed, but only pre-trained models are studied, and not specifically fine-tuned ones. A similar analysis [Gol19] adds more probing tasks and addresses only the BERT architecture.

[QXL+19] focus specifically on analyzing BERT as a Ranking model. The authors probe attention values in different layers and measure performance for representations build from different BERT layers. Like [TXC+19], they only discuss pre-trained models.

There has also been work which studies models not through probing tasks but through qualitative visual analysis. [ZZ18] offer a survey of different approaches, though limited to CNNs. [NSM15] explore phoneme recognition in DNNs by studying single node activations in the task of speech recognition. [HVZ17] go one step further, by not only doing a qualitative analysis, but also training diagnostic classifiers to support their hypotheses. Finally, [LMJ16] take a look at word vectors and the importance of some of their specific dimensions on both sequence tagging

and classification tasks.

The most closely related previous work is proposed by [LGB+19]. Here, the authors also perform a layer-wise analysis of BERT's token representations. However, their work solely focuses on probing pre-trained models and disregards models fine-tuned on downstream tasks. Furthermore, it limits the analysis to the general transferability of the network and does not analyze the specific phases that BERT goes through.

Additionally, our work is motivated by [JW19]. In their paper, the authors argue that attention, at least in some cases, is not well suited to solve the issues of explainability and interpretability. They do so both by constructing adversarial examples and by a comparison with more traditional explainability methods. In supporting this claim, we propose revisiting evaluating hidden states and token representations instead.

## 3.3 Methodology

We focus our analysis on fine-tuned BERT models. In order to understand which transformations the models apply to the tokens we take two approaches: First, we analyze the transformed token vectors *qualitatively* by examining their positions in vector space. Second, we probe their language abilities on QA-related tasks to examine our results *quantitatively*. With this approach we aim for a holistic analysis of the stages found in BERT's transformations, in order to comprehensively answer our research question.

### 3.3.1 Analysis of Transformed Tokens

The architecture of BERT and Transformer networks in general allows us to follow the transformations of each token throughout the network. We use this characteristic for an analysis of the changes that are being made to the tokens' representations in every layer.

We use the following approach for a qualitative analysis of these transformations: We randomly select both correctly and falsely predicted samples from the test set of the respective dataset. For these samples we collect the hidden states from each layer while removing any padding.

This results in the representation of each token throughout the model's layers. Of note here is that we analyze each encoder block as a whole, and not the attention weights as is more commonly done.

The model can transform the vector space freely throughout its layers, and we do not have references for semantic meanings of positions within these vector spaces. Therefore, we consider distances between token vectors as indication for semantic relations.

**Dimensionality Reduction**. BERT's pre-trained models use vector dimensions of 1024 (large model) and 512 (base model). In order to visualize relations between tokens, we apply dimensionality reduction and fit the vectors into two-dimensional space. To that end we apply T-distributed Stochastic Neighbor Embedding (t-SNE) [Maa09], Principal Component Analysis (PCA) [FRS01] and Independent Component Analysis (ICA) [Com94] to vectors in each layer. As the results of PCA reveal the most distinct clusters for our data, we use it to present our findings.

**K-means Clustering**. In order to verify that clusters in 2D space represent the actual distribution in high-dimensional vector space, we additionally apply a k-means clustering [Llo82]. We choose the number of clusters k in regard to the number of observed clusters in PCA, which vary over layers. The resulting clusters correspond with our observations in 2D space.

### 3.3.2 Probing BERT's Layers

n this study, our objective is to delve deeper into how different transformations within a BERT model impact its performance. To achieve this, we implement a variety of semantic probing tasks, aimed at examining the nature of information retained within the transformed tokens after each layer. Our primary interest is to find out if certain layers are dedicated to particular tasks and understand how the model processes and retains language information throughout its layers.

We employ the concept of Edge Probing, as introduced in [TXC+19], which reinterprets core NLP tasks as classification challenges by focusing on their labeling components. This approach provides a unified method for probing across a diverse range of tasks. From the original paper, we

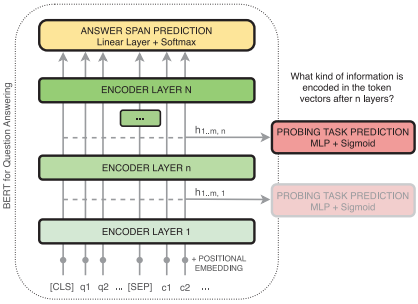



**Figure 3.1.** Schematic overview of the BERT architecture and our probing setup. Question and context tokens are processed by N encoder blocks with a Positional Embedding added beforehand. The output of the last layer is fed into a span prediction head consisting of a Linear Layer and a Softmax Layer. We use the hidden states of each layer as input to a set of probing tasks to examine the encoded information.

adopt Named Entity Labeling, coreference resolution, and Relation Classification, considering their critical roles in language understanding and reasoning [WBC+16]. Additionally, we integrate Question Type Classification and Supporting Fact Identification due to their significance in the context of Question Answering. The source code for our implementation is available at: `https://github.com/bvanaken/explain-BERT-QA`.

**Named Entity Labeling**: This task involves predicting the correct entity category for a given span of tokens. Modeled after Named Entity Recognition but structured as a classification problem, it utilizes annotations from the OntoNotes 5.0 corpus [WHM+11], encompassing 18 entity categories.

**Coreference Resolution**: Here, the model is tasked with determining whether two mentions in a text refer to the same entity. This task, created on top of the OntoNotes corpus and augmented with negative samples by [TXC+19], tests the model's ability to recognize entity references.

**Relation Classification**: The model is challenged to identify the type of relationship connecting two specified entities. Based on data from the

SemEval 2010 Task 8 dataset, which includes English web text and nine directional relation types, this task assesses the model's relational understanding.

**Question Type Classification**: Identifying the type of a question is crucial for answering it accurately. For this task, we use the Question Classification dataset, created by [LR02] from the TREC-10 QA dataset [Voo01]. It contains 500 finely differentiated question types within broader categories such as abbreviation, entity, description, human, location, and numeric value. The entire question serves as the model's input, with its type as the label.

**Supporting Facts**: In the realm of Question Answering tasks, particularly those involving multi-hop reasoning, the ability to extract Supporting Facts is crucial. Our investigation focuses on understanding how BERT's token transformations contribute to identifying key parts of the context in relation to a given question.

To pinpoint the stage at which BERT distinguishes between relevant and irrelevant information, we have developed a probing task centered on identifying Supporting Facts. This task requires the model to determine if a sentence is pertinent to a specific question by containing supporting facts or if it is extraneous. Through this, we aim to test the hypothesis that the token representations within BERT inherently carry information about their relevance to the posed question.

Both the HotpotQA and bAbI datasets provide sentence-level information about Supporting Facts for each question they contain. Since SQuAD does provide such information and requires look at only the sentence containing the answer, we treat the sentence that includes the answer phrase as the Supporting Fact. Additionally, we exclude any QA pairs that have only one context sentence in the example. To assess the model's ability to identify relevant parts specific to each dataset, we have designed distinct probing tasks for each of them. In these tasks, every sentence is labeled as ´true´ if it is a Supporting Fact, or ´false´ if it is not.

**Probing Setup**. Following the approach of the authors in [TXC+19], we

**Table 3.1.** Samples from SQuAD dataset (left) and from Basic Deduction task (#15) of the bAbI dataset (right). Supporting Facts are printed in bold. The SQuAD sample can be solved by word matching and entity resolution, while the bAbI sample requires a logical reasoning step and cannot be solved by simple word matching. Figures in the further analysis will use these examples where applicable.

| | SQuAD | bAbI |
|---|---|---|
| Question | What is a common punishment in the UK and Ireland? | What is Emily afraid of? |
| Answer | **detention** | **cats** |
| Context | **Currently detention is one of the most common punishments in schools in the United States, the UK, Ireland, Singapore and other countries.** It requires the pupil to remain in school at a given time in the school day (such as lunch, recess or after school); or even to attend school on a non-school day, e.g. ´Saturday detention´ held at some schools. During detention, students normally have to sit in a classroom and do work, write lines or a punishment essay, or sit quietly. | **Wolves are afraid of cats.**<br>Sheep are afraid of wolves.<br>Mice are afraid of sheep.<br>Gertrude is a mouse.<br>Jessica is a mouse.<br>**Emily is a wolf.**<br>Cats are afraid of sheep.<br>Winona is a wolf. |

use our fine-tuned BERT model to embed input tokens for each sample across all probing tasks. However, diverging from prior studies, we do this for every layer of the model, employing $N = 12$ layers for BERT-base and $N = 24$ layers for BERT-large. In this setup, we utilize the output embedding exclusively from the $n$-th layer at the $n$-th step. The methodology of Edge Probing stipulates that only the tokens belonging to ´labeled edges´ within a sample are used for classification purposes. For example, in Relation Classification, this would mean considering only the tokens of the two entities in question. These tokens are initially pooled into a fixed-length representation and then processed through a two-layer Multi-layer Perceptron (MLP) classifier. This classifier is responsible for

predicting the probability scores for each label, such as the different types of relations.

For visual clarity, we present a schematic diagram of this setup in Figure 3.1. Additionally, to determine the innate capabilities of the model and differentiate them from the skills it acquires during training, we replicate this probing on pre-trained BERT-base and BERT-large models without any fine-tuning.

## 3.4 Datasets and Models

### 3.4.1 Datasets

In this chapter, our objective is to understand how BERT operates when tackling complex downstream tasks. Question Answering (QA) represents one example of such tasks, necessitating the integration of various simpler operations like coreference resolution and Named Entity Recognition in order to accurately derive answers.

To ensure a comprehensive analysis, we have selected three distinct Question Answering datasets, each with its unique characteristics and challenges. These include SQUAD [RZL+16], a simple real world benchmark, bAbI [WBC+16], which is an artificially generated benchmark designed for more controlled model testing, and HotpotQA [YQZ+18], the most challenging and complex of the three, requiring multi-hop reasoning and the understanding of large contexts. This choice of diverse datasets should provide a comprehensive view of BERT's functionality in QA contexts.

**SQuAD**. The SQuAD dataset, widely recognized as one of the most prominent QA tasks, comprises approximately 100,000 natural question-answer pairs based on around 500 Wikipedia articles. A subsequent iteration, We utilize the version SQuAD 1.1, focusing our attention on the fundamental task of span prediction. Notably, in 2018, an ensemble of BERT models fine-tuned for this dataset achieved a milestone by surpassing the human baseline performance. Conducting our analysis on a dataset where the model exhibits such strong performance, should paint a clear picture of how it works.

**Table 3.2.** Results from fine-tuning BERT on QA tasks. Baselines are: BIDAF [SKF+] for SQuAD, the LSTM Baseline for bAbI from [WBC+16] and the HotpotQA baseline from [YQZ+18] for the two Hotpot tasks.

| | 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 |

**HotpotQA**. The HotpotQA dataset, designed as a Multi-hop QA span prediction task, includes around 112,000 natural question-answer pairs. Unique to this dataset, the questions are crafted to necessitate synthesizing information from multiple sections within a given context. In our study, we concentrate on the ´distractor´ version of HotpotQA, where each context is a blend of relevant and irrelevant facts, averaging about 900 words in length. Given that the pre-trained BERT model has an input limitation of 512 tokens, we reduce the number of distracting facts from the dataset in order to fit within this constraint. Additionally, we exclude the yes/no questions, which constitute about 7% of the dataset. This exclusion is due to their need for a specific architecture, which could potentially skew the results of our analyses. Even with these slight simplifications, BERT is expected to struggle with this dataset, giving us insights as to how BERT behaves differently as uncertainty and errors increase.

**bAbI**. The bAbI QA tasks represent a collection of synthetic tasks, created specifically to probe the capabilities of neural models. Encompassing 20 different tasks, they challenge models to perform Multi-hop QA, requiring reasoning across multiple sentences. These tasks are designed to test a variety of skills, including Positional Reasoning, Argument Relation Extraction, and coreference resolution. Distinct from other QA tasks, the bAbI tasks are characterized by their simplicity, such as a limited vocabulary of around 230 words and brief contexts, as well as the artificial construction of the sentences.

### 3.4.2 Models

In this section we briefly discuss the models our analysis is based on, BERT [DCL+19] and GPT-2 [RWC+19]. Both of these models are Transformers that extend and improve on a number of different recent ideas. These include previous Transformer models [VSP+17][Rad18], Semi-Supervised Sequence Learning [DL15], ELMo [PNI+18] and ULMFit [HR18]. Both have a similar architecture, and they each represent one half of the original Encoder-Decoder Transformer [VSP+17]. While GPT-2, like its predecessor, consists of only the decoder half, BERT uses a bidirectional variant of the original encoder. Each consists of numerous Transformer blocks (12 for small GPT-2 and BERT-base, 24 for BERT-large), that in turn consist of a Self-Attention module, Feed Forward network, Layer Normalization and Dropout. On top of these encoder stacks we add a Sequence Classification head for the bAbI dataset and a Span Prediction head for the other datasets. Figure 3.1 depicts how these models integrate into our probing setup.

### 3.4.3 Applying BERT to Question Answering

We base our training code on the PyTorch implementation of BERT available at [Hug18]. We use the publicly available pre-trained BERT models for our experiments. In particular, we study the monolingual models *BERT-base-uncased* and *BERT-large*. For GPT-2 the small model (117M Parameters) is used, as a larger model has not yet been released. However, we do not apply these models directly, and instead fine-tune them on each of our datasets.

**Training Modalities**. Regarding hyperparameters, we tune the learning rate, batch size and learning rate scheduling according to a grid search and train each model for 5 epochs with evaluations on the development set every 1000 iterations. We then select the model of the best evaluation for further analysis. The input length chosen is 384 tokens for the bAbI and SQuAD tasks and the maximum of 512 tokens permitted by the pre-trained models' positional embedding for the HotpotQA tasks. For bAbI we evaluate both models that are trained on a single bAbI task and also a

multi-task model, that was trained on the data of all 20 tasks. We further distinguish between two settings: Span prediction, which we include for better comparison with the other datasets, and Sequence Classification, which is the more common approach to bAbI. In order to make span prediction work, we append all possible answers to the end of the base context, since not all answers can be found in the context by default. For HotpotQA, we also distinguish between two tasks. In the *HotpotQA Support Only* (SP) task, we use only the sentences labeled as Supporting Facts as the question context. This simplifies the task, but more importantly it reduces context length and increases our ability to distinguish token vectors. Our *HotpotQA Distractor* task is closer to the original HotpotQA task. It includes distracting sentences in the context, but only enough to not exceed the 512 token limit.

## 3.5 Experiments and Results

**Question Answering Performance**. The evaluation results of our top-performing models are summarized in Table 3.2. It is important to note, that the objective of the research presented in this chapter is not to achieve particularly high performance on these datasets, or even compare at all to other baselines. Instead, we depict the performance here to highlight that the models achieve high enough performance for an analysis to be worthwile. Additionally, this helps us put into context some of the analysis results.

The accuracy achieved on the SQuAD task approaches that of human performance, suggesting the model's capability to effectively execute all the sub-tasks necessary for answering questions from the SQuAD dataset. As anticipated, the tasks originating from HotpotQA presented a much greater challenge with its requirement of following multiple steps of reasoning, and finding multiple supporting facts and their relations. Also in line with expectations, both BERT and GPT-2 easily solved the bAbI tasks. Notably, while GPT-2 lagged behind in the more demanding tasks of SQuAD and HotpotQA, it excelled in the bAbi dataset, reducing the validation error to nearly zero.

A closer inspection reveals that the majority of errors BERT encoun-

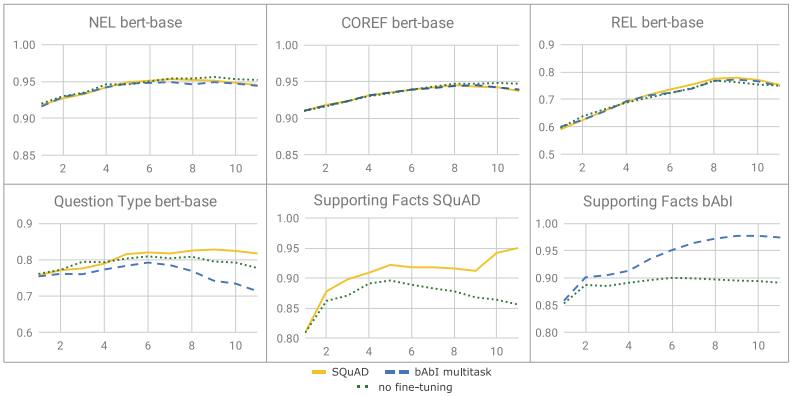



**Figure 3.2.** Probing Task results of BERT-base models in macro averaged F1 (Y-axis) over all layers (X-axis). Fine-tuning barely affects accuracy on NEL, COREF and REL indicating that those tasks are already sufficiently covered by pre-training. Performances on the Question Type task shows its relevancy for solving SQuAD, whereas it is not required for the bAbI tasks and the information is lost.

tered in the bAbI multi-task setting were in tasks 17 and 19, both of which necessitate positional or geometric reasoning. This pattern suggests that GPT-2 might have an edge over BERT in terms of reasoning capabilities in these specific areas.

**Presentation of Analysis Results**. Our qualitative analysis, focusing on the transformations of vector representations, has uncovered several recurring patterns. We will illustrate these patterns using two exemplar samples from the SQuAD and bAbI task datasets, as detailed in Table 3.1.

The outcomes of the probing tasks are visualized in Figures 3.2 and 3.3. In these figures, we present a comparison of macro-averaged F1 scores across all layers of the network. Specifically, Figure 3.2 displays the results from three variations of the 12-layer BERT-base model: one fine-tuned on SQuAD, another on bAbI tasks, and a third without any fine-tuning. Similarly, Figure 3.3 shows the results from two versions of the 24-layer BERT-large model: one fine-tuned on HotpotQA and another without any

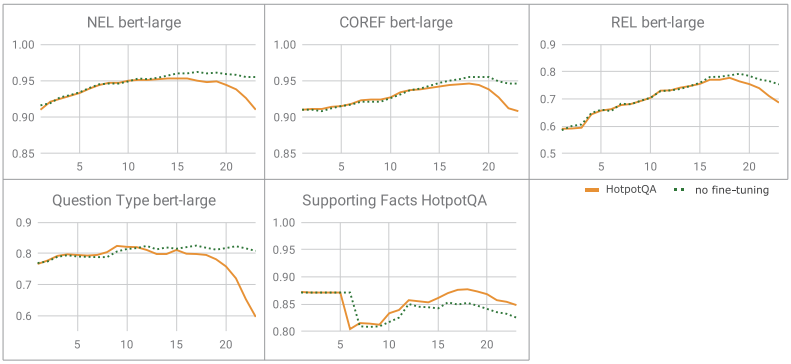


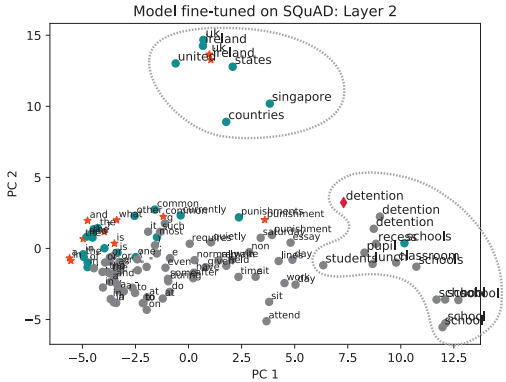

**Figure 3.3.** Probing Task results of BERT-large models in macro averaged F1 (Y-axis) over all layers (X-axis). Performance of HotpotQA model is mostly equal to the model without fine-tuning, but information is dropped in last layers in order to fit the Answer Selection task.

fine-tuning. These comparisons offer insights into the model's performance and transformation patterns under different training conditions.

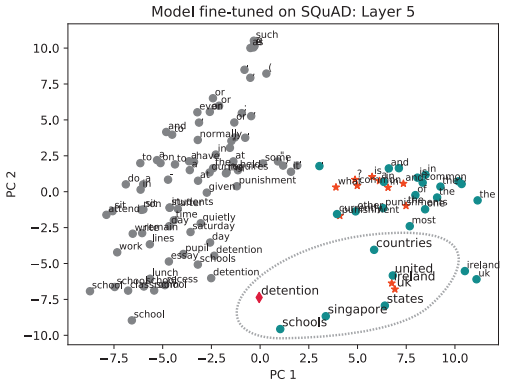
### 3.5.1 Phases of BERT's Transformations

The vector representations of tokens viewed through PCA in different layers suggest that the model is going through multiple phases while answering a question. We observe these phases in all three selected QA tasks despite their diversity in complexity model performance. These findings are further supported by results of the applied probing tasks. We present the four phases in the following paragraphs and describe how our experimental results are linked to our research question.

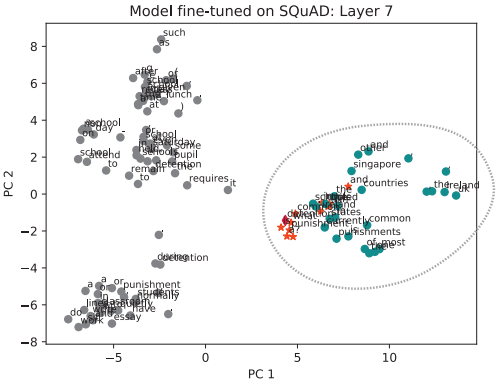
**(1) Semantic Clustering**. Early layers within the BERT-based models group tokens into topical clusters. Figures 3.4a and 3.5a reveal this behaviour and show the second layer of each model. Resulting vector spaces are similar in nature to embedding spaces from e.g. Word2Vec [MCC+13] and hold little task-specific information. Therefore, these initial layers reach



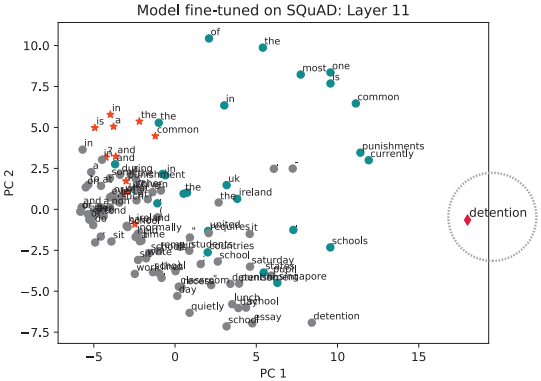

**(a)** SQuAD Phase 1: Semantic Clustering. We observe a topical cluster with ´school´-related and another with ´country´-related tokens.




**(b)** SQuAD Phase 2: Entity Matching. The marked cluster contains matched tokens ´detention´, ´schools´ and the countries that are applying this practice.

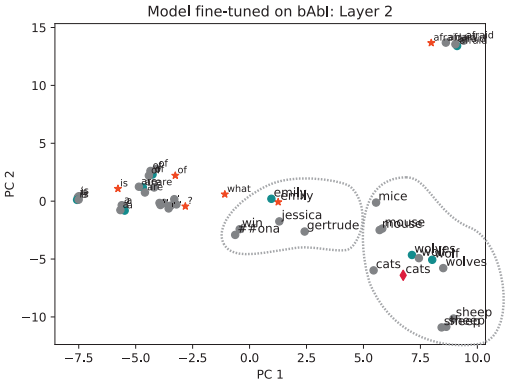



**(c)** SQuAD Phase 3: Question-Fact Matching. The question tokens form a cluster with the Supporting Fact tokens.

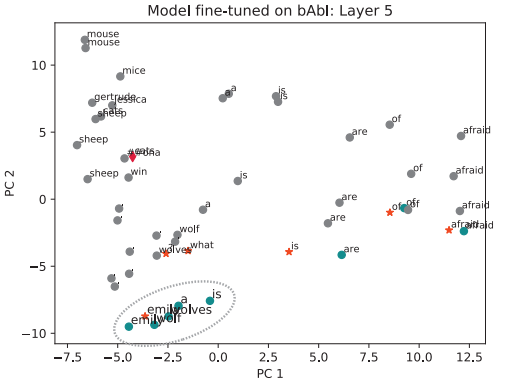



**(d)** SQuAD Phase 4: Answer Extraction. The answer token ´detention´ is separated from other tokens.

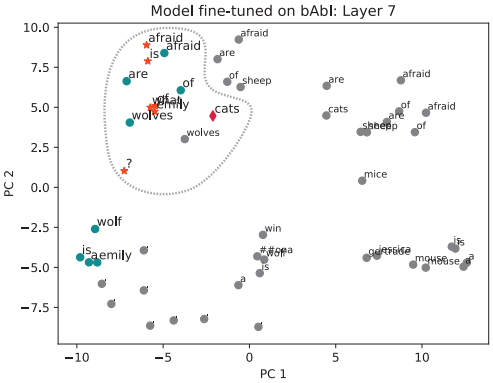
**Figure 3.4.** BERT's Transformation Phases for the SQuAD example from Table 3.1. Answer token: Red diamond-shaped. Question Tokens: Orange star-shaped. Supporting Fact tokens: Dark Cyan. Prominent clusters are circled. The model passes through different phases in order to find the answer token, which is extracted in the last layer (#11).



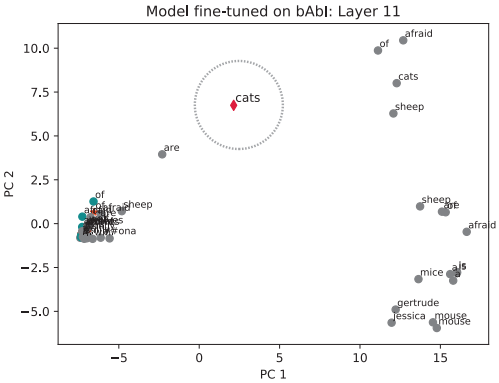

**(a)** bAbI Phase 1: Semantic Clustering. Names and animals are clustered.




**(b)** bAbI Phase 2: Entity Matching. The determining relation between the entities ´Emily´ and ´Wolf´ is resolved in a cluster.




**(c)** bAbI Phase 3: Question-Fact Matching. In this case the question tokens match with a subset of Supporting Facts ('Wolves are afraid of cats'). The subset is decisive of the answer.




**(d)** bAbI Phase 4: Answer Extraction. The answer token ´cats´ is separated from other tokens.

**Figure 3.5.** BERT's Transformation Phases for the bAbI example from Table 3.1. The phases are equal to what we observe in SQuAD and HotpotQA samples: The formed clusters in the first layers show general language abilities, while the last layers are more task-specific.

low accuracy on semantic probing tasks, as shown in Figures 3.2 and 3.3. BERT's early layers can be seen as an implicit replacement of embedding layers common in neural network architectures. With that, they could also be seen as the feature engineering stage from traditional machine learning, as that is what embedding layers have commonly replaced.

**(2) Connecting Entities with Mentions and Attributes**. In the middle layers of the observed networks we see clusters of entities that are less connected by their topical similarity. Rather, they are connected by their relation within a certain input context. These task-specific clusters appear to already include a filtering of question-relevant entities. Figure 3.4b shows a cluster with words like *countries, schools, detention* and country names, in which ´detention´ is a common practice in schools. This cluster helps to solve the question *´What is a common punishment in the UK and Ireland? ´*. Another question-related cluster is shown in Figure 3.5b. The main challenge within this sample is to identify the two facts that *Emily is a wolf* and *Wolves are afraid of cats*. The highlighted cluster implies that *Emily* has been recognized as a relevant entity that holds a relation to the entity *Wolf*. The cluster also contains similar entity mentions e.g. the plural form *Wolves*. We observe analogous clusters in the HotpotQA model, which includes more cases of coreferences.

The probing results support these observations. The model's ability to recognize entities (Named Entity Labeling), to identify their mentions (Coreference Resolution) and to find relations (Relation Recognition) improves until higher network layers. Figure 3.6 visualizes these abilities. Information about Named Entities is learned first, whereas recognizing coreferences or relations are more difficult tasks and require input from additional layers until the model's performance peaks. These patterns are equally observed in the results from BERT-base models and BERT-large models. This maps neatly to the next two stages of the traditional NLP pipeline, named entity recognition, and coreference resolution.

**(3) Matching Questions with Supporting Facts**. Identifying relevant parts of the context is crucial for QA and Information Retrieval in general. In traditional pipeline models this step is often achieved by filtering context parts based on their similarity to the question [JM09]. We observe that

BERT models perform a comparable step by transforming the tokens so that question tokens are matched onto relevant context tokens. Figures 3.4c and 3.5c show two examples in which the model transforms the token representation of question and Supporting Facts into the same area of the vector space. Some samples show this behaviour in lower layers. However, results from our probing tasks show that the models hold the strongest ability to distinguish relevant from irrelevant information wrt. the question in their higher layers. Figure 3.2 demonstrates how the performance for this task increases over successive layers for SQuAD and bAbI. Performance of the fine-tuned HotpotQA model in Figure 3.3 is less distinct from the model without fine-tuning and does not reach high accuracy.[1] This inability indicates why the BERT model does not perform well on this dataset as it is not able to identify the correct Supporting Facts. While these layers dont correspond as concretely to steps in the traditional NLP pipeline, they show what can be expected from an NLP downstream task such a model is applied to, in this case Question Answering. This stage is where the transformations will look the most distinctive between different tasks.

Further, the vector representations enable us to tell which facts a model considered important (and therefore matched with the question). This helps retracing decisions and makes the model more transparent. We discuss this in greater detail in a following paragraph.

**(4) Answer Extraction**. In the last network layers we see that the model dissolves most of the previous clusters. Here, the model separates the correct answer tokens, and sometimes other possible candidates, from the rest of the tokens. The remaining tokens form one or multiple homogeneous clusters. The vector representation at this point is largely task-specific and learned during fine-tuning. This becomes visible through the performance drop in general NLP probing tasks, visualized in Figure 3.6. We especially observe this loss of information in last-layer representations in the large BERT-model fine-tuned on HotpotQA, as shown in Figure 3.3. While the model without fine-tuning still performs well on tasks like NEL or COREF, the fine-tuned model loses this ability. This stage maps to the very end of

[1] Note that the model only predicts the majority class in the first five layers and thereby reaches a decent accuracy without really solving the task.

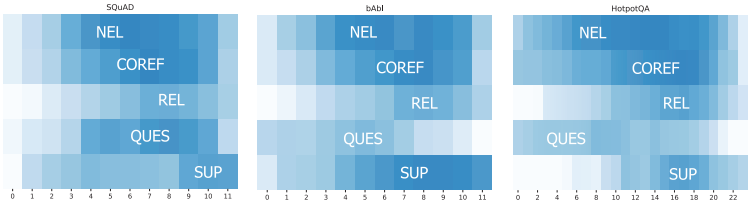



**Figure 3.6.** Phases of BERT's language abilities. Higher saturation denotes higher accuracy on probing tasks. Values are normalized over tasks on the Y-axis. X-axis depicts layers of BERT. NEL: Named Entity Labeling, COREF: Coreference Resolution, REL: Relation Classification, QUES: Question Type Classification, SUP: Supporting Fact Extraction. All three tasks exhibit similar patterns, except from QUES, which is solved earlier by the HotpotQA model based on BERT-large. NEL is solved first, while performance on COREF and REL peaks in later layers. Distinction of important facts (SUP) happens within the last layers.

the NLP pipeline, where we have a clear result extracted or generated for the task at hand.

**Analogies to Human Reasoning**. The phases of answering questions can be compared to the human reasoning process, including decomposition of input into parts [A Z97]. The first phase of semantic clustering represents our basic knowledge of language and the second phase how a human reader builds relations between parts of the context to connect information needed for answering a question. Separation of important from irrelevant information (phase 3) and grouping of potential answer candidates (phase 4) are also known from human reasoning. However, the order of these steps might differ from the human abstraction. One major difference is that while humans read sequentially, BERT can see all parts of the input at once. Thereby it is able to run multiple processes and phases concurrently depending on the task at hand. Figure 3.6 shows how the tasks overlap during the answering process.

In summary, we detect 4 clear phases that BERT goes through, with strikingly similar results across samples from all three datasets. They are highly consistent and visually distinctive, giving us high confidence that our analysis is accurate and comprehensive.

### 3.5.2 Comparison to GPT-2

In this section we compare our insights from the BERT models to the GPT-2 model. We focus on the qualitative analysis of token representations and leave the application of probing tasks for future work. One major difference between GPT-2's and BERT's hidden states is that GPT-2 seems to give particular attention to the first token of a sequence. While in our QA setup this is often the question word, this also happens in cases where it is not. During dimensionality reduction this results in a separation of two clusters, namely the first token and all the rest. This problem holds true for all layers of GPT-2 except for the Embedding Layer, the first Transformer block and the last one. For this reason we mask the first token during dimensionality reduction in further analysis.

Figure 3.7 shows an example of the last layer's hidden state for our bAbI example. Like BERT, GPT-2 also separates the relevant Supporting Facts and the question in the vector space. Additionally, GPT-2 extracts another sentence, which is not a Supporting Fact, but is similar in meaning and semantics. In contrast to BERT, the correct answer ´cats´ is not particularly separated and instead simply left as part of its sentence. These findings in GPT-2 suggest that our analysis extends beyond the BERT architecture and hold true for other Transformer networks as well.

### 3.5.3 Additional Findings

**Observation of Failure States**. One important aspect of explainable Neural Networks is to answer the questions of when, why, and how the network fails. Our visualizations are not only able to show such failure states, but even the rough difficulty of a specific task can be discerned by a glance at the hidden state representations. While for correct predictions the transformations run through the phases discussed in previous sections, for wrong predictions there exist two possibilities: If a candidate answer

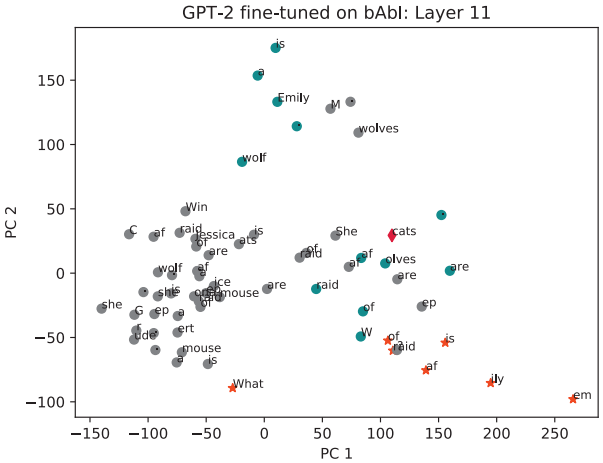



**Figure 3.7.** bAbI Example of the Answer Extraction phase in GPT-2. Both the question and Supporting Fact are extracted, but the correct answer is not fully separated as in BERT's last layers. Also a potential candidate Supporting Fact in ´Sheep are afraid of Wolves´ is separated as well.

was found that the network has a reasonable amount of confidence in, the phases will look very similar to a correct prediction, but now centering on the wrong answer. Inspecting early layers in this case can give insights towards the reason why the wrong candidate was chosen, e.g. wrong Supporting Fact selected, misresolution of coreferences etc. An example of this is shown in Figure 3.8, where a wrong answer is based on the fact that the wrong Supporting Fact was matched with the question in early layers.

If network confidence is low however, which is often the case when the predicted answer is far from the actual answer, the transformations do not go through the phases discussed earlier. The vector space is still transformed in each layer, but tokens are mostly kept in a single homogeneous cluster. In some cases, especially when the confidence of the network is low, the network maintains Phase (1), 'Semantic Clustering'

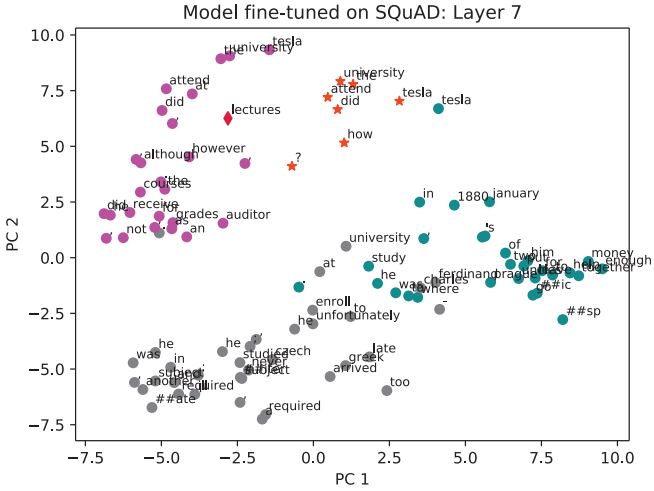



**Figure 3.8.** BERT SQuAD example of a falsely selected answer based on the matching of the wrong Supporting Fact. The predicted answer 'lectures' is matched onto the question as a part of this incorrect fact (magenta), while the actual Supporting Fact (cyan) is not particularly separated.

analogue to Word2Vec, even in later layers. An example is depicted in the supplementary material.

**Impact of Fine-tuning**. Figures 3.2 and 3.3 show how little impact fine-tuning has on the core NLP abilities of the model. The pre-trained model already holds sufficient information about words and their relations, which is the reason it works well in multiple downstream tasks. Fine-tuning only applies small weight changes and forces the model to forget some information in order to fit specific tasks. However, the model does not forget much of the previously learned encoding when fitting the QA task, which indicates why the Transfer Learning approach proves successful.

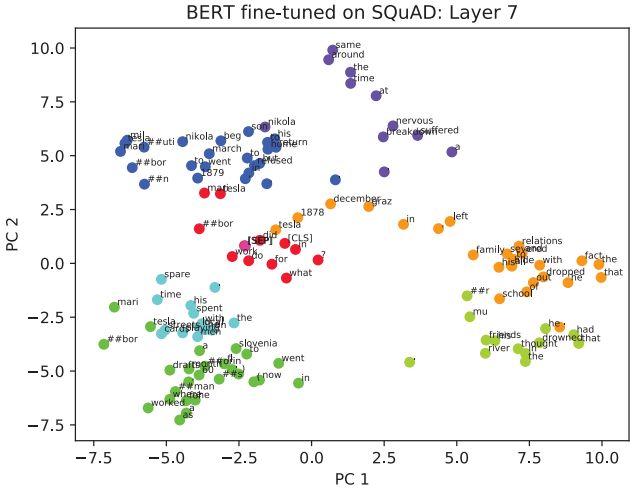



**Figure 3.9.** BERT SQuAD example Layer 7. Tokens are color-coded by sentence. This visualization shows that tokens are clustered by their original sentence membership suggesting far reaching importance of the positional embedding.

**Maintained Positional Embedding**. It is well known that the positional embedding is a very important factor in the performance of Transformer networks. It solves one major problem that Transformers have in comparison with RNNs, that they lack sequential information [VSP+17]. Our visualizations support this importance and show that even though the positional embedding is only added once before the first layer, its effects are maintained even into very late layers depending on the task. Figure 3.9 demonstrates this behavior on the SQuAD dataset.

**Abilities to resolve Question Type**. The performance curves regarding the Question Type probing task illustrate another interesting result. Figure 3.2 demonstrates that the model fine-tuned on SQuAD outperforms the base model from layer 5 onwards. This indicates the relevancy of resolving the

question type for the SQuAD task, which leads to an improved ability after fine-tuning. The opposite is the case for the model fine-tuned on the bAbI tasks, which loses part of its ability to distinguish question types during fine-tuning. This is likely caused by the static structure of bAbI samples, in which the answer candidates can be recognized by sentence structure and occurring word patterns rather than by the question type. Surprisingly, we see that the model fine-tuned on HotpotQA does not outperform the model without fine-tuning in Figure 3.3. Both models can solve the task in earlier layers, which suggests that the ability to recognize question types is pre-trained in BERT-large.

## 3.6 VisBERT

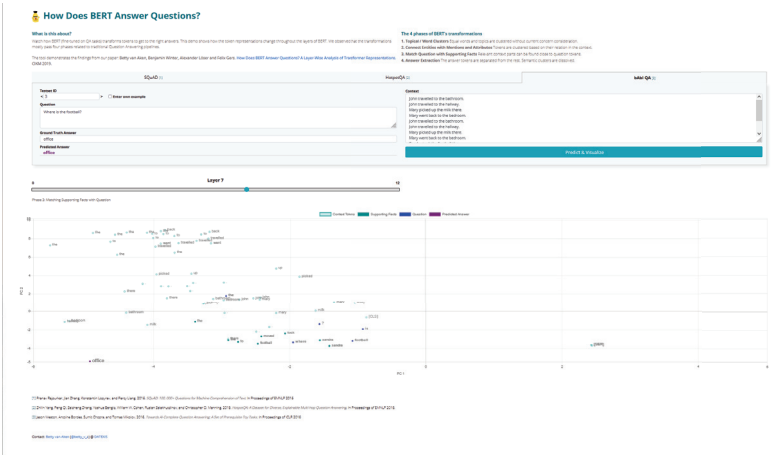



**Figure 3.10.** VisBERT interface. Top: Basic information and data entry. Question, Ground Truth Answer, and question answers are shown and can be edited. Predicted answer by the model is shown as well. Bottom: Hidden state analysis with PCA. Slider controls which layer is shown.

During the work on "How Does BERT Answer Questions? A Layer-Wise Analysis of Transformer Representations"[AWL+19] we further developed

a demonstrator published at WWW2020 and which is available at https://visbert.demo.datexis.com/.

This demonstrator allows the user to explore the qualitative analysis of three different BERT models, trained on SQuAD, HotpotQA and babQA respectively, much in the same way that we have done in the original paper. After choosing either a test set sample from one of the three datasets, or entering a custom question-answering problem, the example is run through a forward pass of the chosen model, and PCA is applied to the output of every hidden layer. Figure 3.10 depicts an overview of the VisBERT interface with one hidden state decomposition. It also showcases one example analysis for the bAbI dataset. This particular case examines Layer 7 of 12 and highlights how tokens of the question (blue) are matched in the same space as the tokens of the supporting facts (dark teal) in the bottom of the chart.

## 3.7 Limitations

While our research offers an in-depth view into how transformer language models process and represent text, the scope of this research is limited. Specifically, we focussed our analysis only on the downstream task of Question Answering. While the early to middle layers of the networks should behave very similarly between downstream tasks, additional insights may be possible when doing a similar analysis for other downstream tasks. Further, base our findings on mainly the BERT family of transformer models, the by far most popular family of models at the time the research was conducted. Our experiments on a GPT model do show similar results, but this might not necessarily hold true for models with different architectures or training schemes.

## 3.8 Summary

In this chapter we performed quantitative and qualitative analyses of transformer language models, addressing research question 1. The qualitative analysis featured dimensionality reduction and an inspection of different words in the vector space, and the quantitative analysis consisted

of performing a variety of probing tasks that highlight different parts of the traditional NLP pipeline. Our work revealed important findings about the inner functioning of Transformer networks in three key areas.

**Interpretability**. The qualitative analysis of token vectors revealed that there is indeed interpretable information stored within the hidden states of Transformer models. Our analysis also provided clues about which parts of the context the model considered important for answering a question. Both the qualitative and quantitative analyses were able to give affirmative evidence to our research question. We showed that the deeply stacked hidden layers of BERT bear a striking resemblance to the traditional NLP pipeline in their functionality. Early layers focus on embedding and simple semantic connections, followed layers proficient in named entity recognition and coreference resolution, then task specific layers, and finally layers that extract and process the actual result.

**Transferability**. We further showed that lower layers might be more applicable to certain problems than later ones. For a Transfer-Learning task, this means layer depth should be chosen individually depending on the task at hand.

**Modularity**. Our findings support the hypothesis that not only do different phases exist in Transformer networks, but that specific layers seem to solve different problems. This hints at a modularity that can potentially be exploited in the training process. This informs our subsequent research in the following chapter.

Finally, we developed an openly available demonstrator which offers the opportunity to perform the same qualitative analysis we did in our research.

Now that we have a deeper understanding of how this architecture transforms data, and have addressed one key challenge in the application of transformer language models in critical scenarios, the next two chapters will address the further challenge of efficient transfer learning with limited data. With the help of the knowledge we have gained about transformers in this chapter, in particular the modularity, we will next present a medical

transformer model with a modular and targeted training approach.

Chapter 4

# Efficiently Integrating Structured Knowledge Into Generic Transformer Models

## 4.1 Introduction

There are two significant parts to answering the research question ´Can over parameterized models be improved through Knowledge Graph Completion Retraining?´. The first part is about verifying that commonly applied transformer models are indeed over parameterized in regard to the downstream tasks we care about. For this thesis we focus on the medical domain for that question, and here specifically information retrieval tasks as well as challenging classification tasks. The second part to answering the research question is developing and evaluating a method to reutilize the superfluous parameters to improve the model in these same downstream tasks. This chapter describes our approaches to both of these parts.

Due to the general nature of pre-training data, transformer models often lack specific domain knowledge or vocabulary and under-perform in even broad domains like the medical one [LYK+20]. While it is possible to pre-train such models on domain specific data, it requires massive amounts of data and computational power. This might not be tractable for certain niche domains and applications. What can be done instead is to make use of the powerful and costly generic pre-training, and impart domain-specific knowledge after the fact. This can reduce data requirements by a large factor. One option to impart this knowledge is to use structured data in the form of knowledge graphs and knowledge bases. These are

attractive knowledge sources since they are widely available in the form of databases often even in niche domains, and they are complementary to the pre-training since that makes use of unstructured data.

The second attribute of these models we exploit in the research presented in this chapter is the fact that they are vastly over parametrized and redundant, as shown by research in the model compression field[MLN19; SDC+19]. We propose KIMERA, a novel re-training method for effective knowledge injection in transformer models which enhances these redundant parameters with the help of structured domain knowledge.

In KIMERA we first detect the redundant attention heads in these transformer models, by using a model compression algorithm. This allows KIMERA to leave the relevant components of the model untouched while improving the more irrelevant ones. We retrain and specialize these redundant components in a Multi-Task training scheme enabling the model to abstract information from the structured knowledge sources. We use common tasks from the Knowledge Graph Completion field to facilitate this training.

We choose Clinical Answer Passage Retrieval(CAPR) and Clinical Outcome Prediction(COP) as downstream tasks. Medical knowledge graphs like UMLS [Bod04] contain commonly known medical knowledge like disease-symptom or drug interactions, while clinical notes often represent the current health state of a particular patient. Therefore, both can effectively complement each other for a deep patient representation. Additionally, we probe our models with GLUE [WSM+19] to assess the effect on the general language abilities that KIMERA retains after the domain transfer. We evaluate the effects of KIMERA on BERT and BioBERT [LYK+20]. BioBERT serves as a strong baseline that is trained with medical data, and our method manages to further improve on its results.The contributions discussed in this chapter and published as [WRL+22] are the following:

- Applying model compression-based analysis for targeted retraining of attention heads
- A novel Multi-Task retraining scheme based on Knowledge Graph Completion to integrate structured knowledge
- Experiments on 5 different strategies to employ our method

- ▻ An evaluation on domain adaptation to the medical domain in 8 downstream tasks over both BERT-base and BioBERT
- ▻ We publish PyTorch code[1] and plan to upload trained models to `huggingface.co`

The remainder of this chapter is structured as follows: Section 4.3 illustrates KIMERA's process; 4.4 introduces the downstream tasks and Knowledge Graphs that we use in our experiments, Section 4.5 discusses the experiments and results on these tasks, Section 4.6 contains an analysis on the actual impact the retraining has on the model, Section 4.2 showcases related work and finally Section 4.8 discusses future work and conclusions.

## 4.2 Related Work

We review here the research that builds the foundation specifically for the KIMERA approach, as well as research that is similar in nature, in order to position this research in the context of the field. This builds on the related work already discussed in Section 2. Most prominently KIMERA is based on research in *Model Compression* and *Knowledge Graphs*. We exclude research on graph neural networks themselves from this short survey, as these models are altogether different in spirit, architecture and purpose from the KIMERA approach.

### Model Compression

The goal of KIMERA is not only to retrain a model with domain specific data, but to do so in a targeted manner to combat both over parametrization and catastrophic forgetting. To that end we need to identify those parts of the underlying model that are superfluous, either generally or given a specific downstream task. This is the exact aim of model compression techniques as well. However, where model compression utilizes such information to remove the superfluous parameters, and thereby create smaller and faster models, we seek to instead reuse and repurpose them to create a more powerful model of same size as the original.

[1] https://anonymous.4open.science/r/kg-transformers/README.md

4. Efficiently Integrating Structured Knowledge Into Generic Transformer Models

There are two major groups of model compression techniques: pruning, where certain model parameters are removed directly within the model [SLM16], and Knowledge distillation or student-teacher, where the bigger original model ´teaches´ a separate, smaller model[SDC+19]. Pruning is the approach more useful to our research, since it helps us analyze and evaluate models in and of themselves. Specifically, we follow [MLN19] to calculate the importance of attention heads in our model as described in 4.3. Authors of [MLN19] demonstrate that using their approach a substantial amount of attention heads can be removed from the network with only minimal impact to downstream task performance.

## Knowledge Graphs

When integrating knowledge graph information in non-graph neural networks there are two options. The first is *Structured Knowledge Integration*, which queries the knowledge graph or sub graphs explicitly ([ZXQ+20], [BC19], [LZZ+20], [ZWM19]). For transformers specifically, there are *Adapter*-based approaches that train small modules on top of the existing model [PNI+19; HZX+20; WTD+20]. KIMERA's aim however is to address over parametrization, not cause more of it. To keep the model efficient we also want to avoid having to query external knowledge sources after training is done. Therefore, these approaches are of little help to our research.

The second category of approaches to integrating knowledge graph information is *Knowledge Injection*. KIMERA is fits this category as Knowledge injection is concerned with implicitly introducing additional knowledge during the (pre-) training process. [FDJ+15] for example match the euclidean distance of pairs of their word vectors, to the distance of the words in a knowledge graph, leading to more semantically powerful word vectors. [YCW+19] develop a novel pre-training strategy using multiple-choice questions derived from a commonsense knowledge graph. [ZDW20] utilize a pre-trained BiLSTM and train it on a Concept Alignment task based on UMLS. [WGZ+21] and [HZP20] on the other hand add additional objectives during pre-training in the form of knowledge embedding and concept relation prediction respectively. Most closely related to KIMERA, [KHK+20] follow a multi-task knowledge graph completion

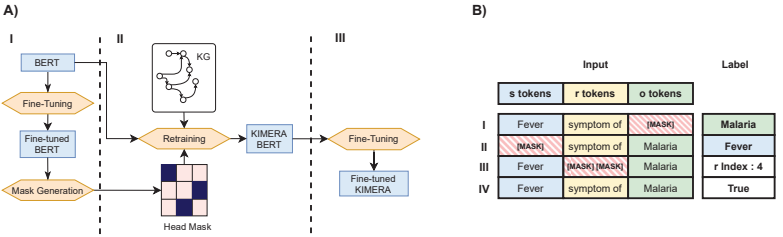



**Figure 4.1. A)** KIMERA consists of three phases: **I** A transformer model is fine-tuned and a head-mask is computed by identifying redundancies. **II** The computed mask is then used in conjunction with a multi-task training based on knowledge graph completion. Finally, the model is fine-tuned on the target task. **III** The retrained model is fine-tuned on the domain-specific task to culminate the domain transfer. **B)** Examples of KG retraining tasks. **I** and **II** *Entity Prediction* with a Masked Language Modelling objective. **III** *Relation Prediction* with a multi-class classification objective, and **IV** *Triplet Classification* with a binary classification objective.

approach to improve a pre-trained transformer model. However, we differ in both the actual knowledge graph completion tasks chosen, and in targeting only specific attention heads with our retraining.

We choose *Knowledge Graph Generation* in particular because of the implicit and straightforward nature of the task, because these tasks are closely aligned with the pre-training that transformer language models already received, and because transformers have been shown very adept at this set of tasks [PRR+19; YML19; BRS+19]. We build on these works by adding and additional triplet verification task, and using them as an intermediary step in our retraining scheme.

## 4.3 Methodology

Requirements to apply our method are a pre-trained language model, and a (domain-specific) knowledge base or graph. In particular, the knowledge source should be chosen in a way that its contained information and language is both lacked by the pre-trained language model, and is useful

for the downstream tasks that are to be solved by the transfer learned model. Optimally choosing or creating this knowledge graph is not a trivial task by itself, a topic which requires further research. Our KIMERA approach then consists of three steps which are outlined in Figure 4.1 **A)**.

## Step I: Importance Approximation

The first step consists of the computation of importance scores $I_h$ for the attention heads of our model. This will inform our targeted retraining. To that end we first fine-tune the pretrained transformer model on a target downstream task. This lets us calculate Importance scores by following [MLN19]. They make use of a modified version of multi-headed attention $MHAtt$[VSP+17] with an additional, per-head, binary flag $\xi_h$

$$MHAtt(\mathbf{x}, q) = \sum_{h=1}^{N_h} \xi_h Att(\mathbf{x}, q) \tag{4.3.1}$$

$\xi_h$ simply allows singular heads to be turned on or off dynamically. [MLN19] then understand the importance of an attention head as the expected sensitivity of the downstream task loss $\mathcal{L}(x)$ to having that head turned on or off. The higher the discrepancy in the losses, the more important the head is. This results in the following equation:

$$I_h = \mathbb{E}_{x \sim X} \left| \frac{\partial \mathcal{L}(x)}{\partial \xi_h} \right| \tag{4.3.2}$$

This $I_h$ can also be approximated by collecting gradients during a training epoch on the downstream task, or even on just a small set of samples. In practice [MLN19] suggest applying this iteratively, turning off a fraction $\rho$ heads with the smallest gradients per step, until downstream task performance falls under a certain threshold $\tau$. This process finally leaves us with a pruning mask $M_{hard}$ of zeroes and ones, describing which heads where turned on and off at the end of the process. For KIMERA we utilize $M_{hard}$ in a way where a 1 denotes an attention-head that is already relevant to the downstream task and should be kept as is, and a 0 denotes an attention head that is superfluous, and which can be repurposed with additional retraining. This is where one of our main contribution lies, reusing those redundant heads instead of discarding them.

When creating $M_{hard}$ in this way for multiple downstream tasks, either one mask per task can be created, or one one singular mask via multi-task training during importance score calculations. The former strategy is more accurate but necessitates multiple retraining and fine-tuning steps, making this strategy less efficient. It does however result in a number of more powerful models equal to the number of tasks.

## Step II.I: Applying the Masks

In order to make use of the $M_{hard}$ we just calculated and target the coming retraining to only redundant heads, we modify the parameter updates in the following way:

$$W_{i+1}^{lh} = W_i^{lh} - \eta(1 - m^{lh})\nabla\mathcal{L} \tag{4.3.3}$$

with $W_i^{lh}$ being one of the $(Q, K, V, O)$ matrices[VSP+17] for the attention head $h$ in layer $l$ at training iteration $i$. $\nabla\mathcal{L}$ denotes the loss gradient and $\eta$ is the general learning rate and $m^{lh}$ is the mask value of head $h$ at layer $l$. In effect, we scale the learning rate by value of the mask for each head.

We evaluate 3 different strategies to choosing and applying this mask:
**Discrete learning rate adaptation**. The most basic application of our method makes use of the hard mask $M_{hard}$. Here, redundant heads train with an unmodified learning rate during the training, while heads that survived the importance calculation with a 1 have their learning rate set to 0 and thereby do not change at all. We call them frozen. Since heads in a transformer layer interact with each other to reach the combined next hidden state. Partially freezing specific heads during the retraining, might make it too challenging for the network to yield a coherent representation using all heads for the eventual down stream task. We thus explore an alternative method to not freeze attention heads completely, but rather weight the learning rate smoothly:

**Soft attention-head mask**. For this we require a smooth mask, rather than the binary $M_{hard}$. We modify the importance score calculations of [MLN19] such that when an attention would be turned off, namely its $\xi_h$ set to 0, it is instead set to the last normalized value of $I_h$. The

resulting mask $M_{soft}$ contains an approximate Importance value for each head. We use this instead of $M_{hard}$ and apply it in the same way during retraining. In this strategy no heads are completely frozen, but with the inverse weighing of the learning rate in respect to the importance less important heads are changed more drastically, and important heads only mildly.

**Weighing the forward pass**. Our first 2 strategies only apply to the backward pass and modify the learning itself. We further want to evaluate whether there is merit in isolating the forward pass as well. Therefore in this third strategy we apply the same mask both in the forward and backward passes. This further limits the influence that the important attention heads have during the retraining.

## Step II.II: Retraining

Now equipped with a mask that separates the important and unimportant, or necessary and superfluous parameters, the next stage is to develop the method improve these superfluous parameters. This method is detailed henceforth. It uses a pre-trained model, the attention masks computed in the previous step, and a knowledge graph, resulting in a model that can be fine-tuned on the final downstream task. We follow a multi-task training scheme with tasks based on knowledge graph triplets. We adopt the common Knowledge Graph Completion tasks of *entity prediction*, *relation prediction*, and *triplet classification*, e.g. [BWC+11; SCM+13; YML19], and apply them in this novel way. These tasks are intended to specialize the redundant or unimportant attention heads into the domain of the knowledge base.

**Multitask Training Scheme.** We follow a multi-task scheme to force the target models to generalize by having a combination of multiple competing losses. We explore two different settings. First, we attempt to improve existing pre-trained transformer models, namely BERT or BioBERT, by retraining them. In the second setting, we train BERT from scratch exclusively on the knowledge graph completion tasks to measure the extent of the complementary information added by a knowledge graph. In each task, we target a single knowledge graph triplet denoted in a directed

graph by $(s, r, o)$: subject node, relation edge, and object node, respectively. We adopt three link prediction tasks focusing each on completing one of these $s$, $r$, or $o$ triplet elements, and a fourth task validating the plausibility of the whole triplet. Figure 4.1 **B)** depicts examples for these tasks. Each input row depicted in this figure is embedded as a single input sequence, with separator tokens between the columns.

**Entity Prediction**. We frame entity prediction as a Masked Language Modelling task [DCL+19]. In our multi-task setting, this results in two tasks: given $(s, r)$ or $(r, o)$, $o$ or $s$ have to be generated correspondingly. In contrast to [DCL+19], we mask and predict all tokens of $o$ or $s$. This generation results for both cases in a sequence of tokens denoting the model's predictions for the masked component. The loss being optimized is token-wise cross-entropy over the model vocabulary.

**Relation Prediction**. In this task, given $(s, o)$, the objective is to predict $r$. While this task could also be modeled with a (masked) language modeling objective similar to the Entity Prediction tasks, we opt to implement this task as a multi-class classification since, in our case, the number of relations in the graph is very small compared to BERT's vocabulary. This simplifies the task substantially.

**Triplet Classification**. This task tests if a graph triplet is a valid triplet present in the knowledge graph. Given a triplet $(s, r, o)$, this task involves a binary classification to determine its plausibility. We take valid samples directly from the knowledge graph and generate an equal amount of invalid samples by replacing one of the three components with the same component from a different randomly selected triplet.

**Multitask model architecture**. To implement this multi-task setting we use the encoder part of the transformer model, pool the output, and add linear layers, one for each task. These output layers have the same size as the hidden size of the transformer model used. We experiment with different pooling techniques as hyperparameters, e.g. [CLS] token for BERT, average pooling, max pooling, and a learned pooling method using an additional linear layer.

**Optimization Objective**. During training, we sample batches randomly from all tasks and compute the main loss as a weighted sum of losses corresponding to each one of the tasks

$$\mathcal{L} = \alpha_1 \mathcal{L}_1 + \alpha_2 \mathcal{L}_2 + \cdots + \alpha_n \mathcal{L}_n \tag{4.3.4}$$

where $\alpha_1, \ldots, \alpha_n$ are scalar loss weights which are regarded as hyperparameters, and $\mathcal{L}_1, \ldots, \mathcal{L}_n$ are the per-task loss functions, namely Categorical Cross Entropy in all tasks. This weighted sum over the tasks is to weigh difficult tasks more strongly to prevent overfitting on some of the simpler tasks.

### Step III: Fine-tuning

This is the final step proposed in KIMERA and it involves extracting the encoder from the retrained model and fine-tuning it on the final downstream task as is common practice, yielding a model with specific domain knowledge. During this final fine-tuning no masks or learning rate modifications are applied.

## 4.4 Datasets and Downstream Tasks

Ideally, the knowledge graph that we instill into a language model has large amounts of complementary information and is relevant for solving the downstream task. The performance of our retraining method relies on the combination of *knowledge graph, language model, downstream task* fitting appropriately. We leave metrics and an algorithm for automatically evaluating the fitness of such a combination to future work. To evaluate our method, we choose eight datasets from the clinical domain with challenging tasks such as zero shot-retrieval and extreme multi-class classification on hundreds of classes. The clinical domain in particular exhibits issues like limited training data, due to privacy and regulatory issues, and idiosyncratic language, which may highlight insufficiencies in BERT's capabilities [KS20]. Additionally, there is reasonable structured data available for this domain in the form of UMLS[Bod04]. It is for these

reasons that we decide on the clinical domain to evaluate KIMERA. We specifically highlight the clinical domain, which is closely concerned with direct patient care, as a subset of the general biomedical domain. We choose our tasks in favor of common tasks such as Named Entity Recognition and Relation Extraction since in a clinical setting doctors do not find this type of information extraction sufficient. Instead, they deem complex downstream tasks such as patient cohort retrieval and outcome prediction more useful [MLK16; Top19].

### 4.4.1 Knowledge Graphs

We combine three knowledge graphs into one dataset: UMLS[Bod04], HSDN[ZMB+14], and the graph from [RHT+17]. We gather ∼2.5M knowledge graph triplets with 43 unique relation types. We limit the sequence length of nodes to 100 tokens and edges to 10 tokens, and pad accordingly. This is done to optimize computation speed while truncating $< 0.1\%$ of triplets.

**UMLS**[Bod04] The Unified Medical Language System is an aggregation of different medical knowledge sources. This work specifically focuses on UMLS' Metathesaurus, which contains diseases, symptoms, medications, etc., and the relations between them. From the 80 million relationship triplets in UMLS, we filter for relevant relation types, triplets that are complete, and choose to keep only well-populated sub-relations with more than 10k sample triplets. This results in our training corpus of ∼600k triples.

**HSDN**[ZMB+14] is constructed from ∼7M PubMed[SAB+18] bibliographic records. MeSH(Medical Subject Headings)[LB94] metadata is used to identify symptom and disease terms. The co-occurrence of at least one symptom and one disease term is then utilized to filter the PubMed records further. From these records, symptom-disease relations are then extracted, resulting in ∼150k triplets.

**[RHT+17]** create a knowledge graph from electronic health records collected between 2008 and 2013 from a trauma center and tertiary academic

teaching hospital. Concepts are extracted by applying UMLS as well as other sources to these records. The graph is then constructed by a set of 3 probabilistic models which relate symptoms and diseases. The resulting graph contains ∼3k symptom-disease triplets.

### 4.4.2 Clinical Answer Passage Retrieval(CAPR)

Retrieving documents and passages from clinical documents is an important task in the medical domain. We evaluate our models on the clinical answer passage retrieval task(CAPR) [GAL21] in a *zero-shot setting* and across four different datasets. The zero-shot setting puts an even higher burden on each individual model since each model is evaluated as-is, and not fine-tuned to the evaluated datasets. We follow [GAL21] and evaluate our models using the Cross Encoder Architecture [HSL+20], which calculates matching scores over the joint sequence of all query and passage pairs. We use the same training and evaluation described in [GAL21] and train on Wikipedia articles, and evaluate on WikiSectionQA[AAG+20], Mimic-III clinical notes[JPS+16], MedQuad[AD19], and HealthQA[ZAW+19] datasets. In this setting, we create only one joint attention-head mask for all four tasks. This mask is generated on a dataset that is combined from held out parts of the test sets of each of the datasets.

### 4.4.3 Clinical Outcome Prediction(COP)

We adopt the admission notes dataset by [APM+21] for the Clinical Outcome Prediction tasks. They are based on special filtering of Mimic-III's discharge summaries that simulate patient information at the time of admission. This is achieved by only keeping the following sections: *Chief complaint, (History of) Present illness, Medical history, Admission Medications, Allergies, Physical exam, Family history, Social history.* In particular, this filtering hides all information about the course and outcome of treatment of the patient during their stay. We evaluate 4 different tasks related to clinical outcome of patients, outlined in the following.

**In-hospital Mortality Prediction Task (MP)** This task is a binary classification task, in which the model determines whether a patient deceased during the hospital stay or not. The data is heavily imbalanced with 90% of patients surviving their stay.

**Length of Stay Prediction Task (LOS)** Here the model classifies a patient's stay at the hospital into 4 classes regarding the length of their stay: $< 3$ *days*, $3 - 7$ *days*, $1 - 2$ *weeks*, $2+$ *weeks*.

**Diagnosis Prediction Task (DIA)** In this extreme multi-label classification task the model is tasked with assigning ICD-9 diagnosis codes to a patient. Instead of 4-digit codes, we reduce the problem to 3-digit codes, which results in 1266 ICD-9 codes with a power-law distribution.

**Procedures Prediction Task (PRO)** This task follows the diagnosis prediction task, being a multi-label task utilizing 3-digit ICD-9 codes. There are 711 procedure codes that we use from Mimic-III.

## 4.5 Experiments and Results

Our Experiments and Baselines are based on either BERT-base or BioBERT. Although ClinicalBERT [AMB+19] is another option for comparison, we do not consider it for our evaluation since it is already trained on Mimic-III, skewing the results especially in the zero-shot CAPR scenario.
For BioBERT we choose *dmis-lab/biobert-v1.1* from the huggingface transformers repository [WDS+20], and for BERT-base experiments we choose the best model out of BERT-base-uncased and BERT-base-cased. For the Clinical Answer Passage Retrieval, we find that hyperparameter optimization does not have a significant impact, and manually choose reasonable values from several trials. In contrast, Clinical Outcome Prediction is very sensitive to hyperparameters. Therefore, we carry out a thorough hyperparameter optimization based on HyperOpt [BYC13] for all evaluated models. All KIMERA models are trained on the full set of knowledge graph triplets and for a maximum of 5 epochs, but most models converged after a single epoch. Although the parameter $\alpha$ could weigh partially the

loss on the tasks, in our experiments it was only used discretely to enable or disable distinct tasks. We find in our experiments that it is usually most beneficial to keep all $\alpha_n$ at 1 and leave the exploration of soft weightings to further research. On a single Nvidia V100 GPU, one epoch takes 18 hours. We choose the head masks resulting from the best base model, calculated with performance threshold $\tau \in [0.95, 0.98, 0.99]$ and a per step pruning ratio $\rho = 0.1$. We explore the effect of the selective retraining of attention heads with KIMERA is done in 4.6.

**Table 4.1.** Results across the four CAPR datasets using the Cross Encoder architecture(left) and four COP tasks(right). Top part shows scores for models based on BERT-base, bottom part scores for models on BioBERT. KIMERA improves on both BERT-base and BioBERT performance, with the exception of the LOS task.

| Model | MedQuad | | HealthQA | | Mimic-III | | Wiki | | MP | LOS | DIA | PRO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 | AUROC | AUROC | AUROC | AUROC |
| BERT-base | 52.63 | 60.80 | 40.30 | 81.82 | 59.74 | 72.07 | 35.44 | 77.66 | 81.13 | **70.40** | 82.08 | 85.84 |
| BERT-base(pruned) | 50.71 | 60.45 | 39.92 | 78.12 | 61.96 | 72.64 | 35.23 | 75.12 | 81.07 | 70.14 | 80.21 | 83.48 |
| KIMERA scratch | 32.88 | 74.17 | 31.23 | **83.45** | 23.63 | 41.77 | 20.63 | 59.85 | 75.75 | 65.74 | 51.1 | 64.91 |
| KIMERA no-mask | 64.68 | 92.33 | 49.01 | 80.31 | 65.68 | 79.78 | 50.38 | 80.44 | 81.63 | 69.55 | 82.47 | 85.91 |
| KIMERA hard-mask | **71.94** | **94.52** | **50.53** | 82.71 | 67.13 | 80.52 | **51.73** | 80.72 | **81.88** | 69.02 | **82.59** | **85.95** |
| KIMERA soft-mask | 70.33 | 93.81 | 49.50 | 81.69 | 67.94 | **81.82** | 51.25 | **81.31** | 81.20 | 68.11 | 82.35 | 85.49 |
| KIMERA b+f | 70.41 | 93.91 | 49.22 | 80.99 | **68.07** | 80.43 | 50.81 | 81.24 | 65.72 | 55.36 | 81.45 | 84.21 |
| BioBERT | 78.86 | 97.06 | 62.07 | 91.59 | 64.89 | 78.81 | 61.31 | 90.69 | 82.55 | **71.59** | 82.81 | 86.36 |
| KIMERA BioBERT | **79.74** | **97.93** | **64.14** | **92.26** | **65.22** | **79.02** | **62.48** | **94.32** | **82.87** | 71.42 | **83.56** | **88.44** |

### 4.5.1 Models and Baselines

We focus on the BERT architecture and the domain specific BioBERT, exploring different variations of KIMERA trained from these base models.

**BERT-base** [DCL+19] We focus on the smaller BERT-base and choose from the English pre-trained models and use the best of BERT-base-uncased and BERT-base-cased for each task.

**BERT-base(pruned).** This model is created applying the pruning scheme of [MLN19] to BERT-base. The authors showed that this model sometimes outperforms BERT-Base solely due to pruning. Therefore, we include this baseline to confirm that the improvements of our methods cannot be

achieved solely by pruning.

**BioBERT** [LYK+20] follows the same architecture as BERT-base-cased. This model is a state of the art biomedical language model, and is pre-trained on PubMed for 23 days on 8 V100 GPUs. This is up to $50-250$ times slower than using KIMERA to create a domain-specific model.

**KIMERA no-mask**, **hard-mask**, **soft-mask** make use of different types of masks during the retraining step. *no-mask* uses no mask at all, whereas *hard-mask* and *soft-mask* explore the corresponding discrete and soft learning rate adaptation proposed in 4.3.

**KIMERA from-scratch.** We investigate the KG retraining as the sole pre-training step. We randomly initialize BERT-base apply the multi-task KG training, before fine-tuning on the downstream tasks.

**KIMERA b+f.** We base KIMERA b+f on KIMERA hard-mask, but apply the mask both in the backward and forward pass as discussed in 4.3, which leads to a strict isolation between frozen and unfrozen heads.

**KIMERA BioBERT** follows *KIMERA hard-mask* but uses BioBERT as a base model. Here we probe whether KIMERA can also be used for improving already domain-specific models with additional structured data, besides efficient domain transfer.

### 4.5.2 Clinical Answer Passage Retrieval

We choose to calculate only one joint attention mask ahead of retraining instead of individual ones for each task, due to the zero-shot setting of this benchmark. Table 4.1 reports results in these tasks. The Cross Encoder shows significant performance differences between models. Most notably *KIMERA hard-mask* and *KIMERA soft-mask* outperform BERT-base across all tasks with a margin of up to 20% in R@1 and up to 35% in R@5. Even *KIMERA no-mask* achieves notable performance boosts. This can be ascribed to the functioning domain transfer with the help of information from UMLS. We also evaluate our methodology on BioBERT and manage to overcome it in all the retrieval tasks, suggesting that KIMERA serves as

well to further specialize BioBERT in the medical domain. In the case of Mimic-III, BioBERT is only marginally ahead of BERT-base. KIMERA only beats both of them by a few percentage points, in contrast to the other tasks. One reason for this could be that domain-specific data here is less relevant than for the other tasks.

In general, using an attention-head mask during the re-training does lead to a performance increase over our no-mask approach. However, none of the masking strategies is clearly better than the others. KIMERA from-scratch generally under-performs in all of the retrieval tasks. This reinforces the fact that the information contained in UMLS is only complementary and not a replacement to the general language capabilities of a pre-trained model. Simply pruning the model did also not improve performance for these tasks except Mimic-III. This demonstrates that the performance increases we observe for KIMERA do not stem from the pruning alone. In summary, for this benchmark our method using knowledge graph completion leads to significant improvements in the model.

### 4.5.3 Clinical Outcome Prediction

For this benchmark an attention mask is generated for each of the tasks individually. In contrast to the Passage Retrieval tasks, the COP results show significantly lower variance in the performance between models. [APM+21] highlight numerical errors as one of the major error classes in these tasks, emphasizing that their evaluated models do not follow medical reasoning, but focus on statistical observations. This fact in combination with the already strong performance of the base architecture of BERT-base could account for the small variance.

As shown by Table 4.1, KIMERA BioBERT achieves the best results with the exception of the LOS task. Similarly, when applying KIMERA to BERT-base we achieve consistent improvements. The different masking strategies of KIMERA performed closely without any particular one standing out as the best. The results of KIMERA *from-scratch* confirm the complementary nature of the UMLS data we found also in the Passage Retrieval tasks. The pruned BERT-base model did not provide performance benefits in these tasks either.

For both the *Mortality Prediction* and *Length of Stay* tasks the back + forward approach performed significantly worse. Given the almost equal performance to other KIMERA models in other tasks, we deem these as outliers that are caused by an insufficient amount of hyperparameter optimization.

**Table 4.2.** Results of the GLUE benchmark, choosing the best of 10 seeds. KIMERA consistently outperforms BioBERT, and shows improvements over BERT-base in 3 tasks, having the highest mean score of tested models.

| Model | CoLA | SST-2 | MRPC | STS-B | QQP | MNLI | QNLI | RTE | WNLI | Mean |
|---|---|---|---|---|---|---|---|---|---|---|
| BERT-base | 59.05 | **93.34** | **89.37** | **88.79** | 89.84 | **85.12** | **91.78** | **69.31** | 49.30 | 79.54 |
| BioBERT | 43.70 | 91.28 | 88.51 | 88.15 | 89.59 | 83.97 | 90.84 | 67.50 | 32.39 | 75.10 |
| KIMERA no-mask | 60.17 | 92.20 | 87.71 | 88.12 | 89.53 | 84.49 | 90.35 | 67.50 | 60.17 | 80.02 |
| KIMERA hard-mask | **62.06** | 93.00 | 88.93 | 88.53 | **90.63** | 84.65 | 91.15 | 69.12 | **62.05** | **81.13** |

The LOS stands out as the only downstream task, including the results in CAPR, where KIMERA did not achieve improvements.

In summary, while the differences are not as significant as in the CAPR tasks, KIMERA did lead to improvements for the COP tasks as well, giving further positive evidence for our research question.

### 4.5.4 General Language Understanding (GLUE)

We evaluate KIMERA on GLUE [WSM+19] and compare it to BERT-base and BioBERT. The results are detailed in Table 4.2. KIMERAmodels for this evaluation have been trained on the medical KGs with masks generated in CAPR, in order to assess how the medical transfer learning impacts the language capabilities. As expected, BERT-base outperforms the biomedically trained BioBERT across all tasks with its general language pre-training. Furthermore, the comparison between KIMERA no-mask and KIMERA hard-mask shows that the hard-mask version, where only a subset of the attention heads have been retrained, is consistently superior to the non-mask version. This supports our intuition that the masking process enables the model to retain more of its language ability during the transfer learning process. Notably, KIMERA outperforms even BERT-base in 3 of the GLUE tasks. While we expected KIMERA with clinical training to perform slightly worse than BERT-base since the knowledge graph task data does not contain proper grammar in its triplets and therefore skews language perception, the results show that for CoLA, QQP and WNLI tasks this training is particularly beneficial and leads to significant improvements over BERT-base.

### 4.5.5 Additional Experiment: Common-Sense

The research that inspired the first chapter of this thesis revealed a lack of ´thinking´ and by extension common sense in generic transformer models. Given the existence of commonsense knowledge graphs like Atomic and ConceptNet, it is natural to explore whether KIMERA can be applied to remedy this shortcoming. This section details our experiments on that front.

#### Datasets

We detail here both the knowledge graphs involved in this experiment, and the downstream task benchmark we use to measure the success of this experiment.

**ConceptNet** ConceptNet is an extensive knowledge graph that captures the variety of relationships and meanings among words and phrases in natural language. It achieves this through the use of labeled, weighted edges connecting different terms, providing a structured method to understand language. It was initially born out of the Open Mind Common Sense project, a crowd-sourced knowledge initiative, and has evolved significantly since its first release. We focus specifically on version ConceptNet 5.5, which integrates lexical and world knowledge from a diverse range of sources and languages.

There are a number of different relationships between words that this knowledge graph can model. For example, it can illustrate usage (´An axe is used to split wood´), lexical forms (´trunks´ as a form of the word ´trunk´), as well as translations between languages (´hot´ in English translates to ´ gorący´ in Polish).

It not only provides information about each word but also connects users to external resources such as WordNet, Wiktionary, and DBPedia which offer additional definitions and contexts. With all of these semantic relationships and external links, we expect that ConceptNet is a very rich source of many types of common sense, representing the most basic types of information every human would have about the included words.

4. Efficiently Integrating Structured Knowledge Into Generic Transformer Models

**Atomic** ATOMIC is an atlas structured around 877,000 textual entries detailing everyday common knowledge. This resource primarily emphasizes inferential knowledge, unlike other databases that concentrate on taxonomic details. ATOMIC organizes this information into specific if-then relational types with variables, such as ´if X attacks Y, then Y will likely defend themselves.´ There are nine such distinct if-then relational types to differentiate between causes and effects, agents and themes, voluntary and involuntary events, and actions versus mental states.

Authors of this knowledge graph illustrate that models can develop rudimentary commonsense reasoning abilities when trained on this knowledge graph and effectively handle scenarios they have not previously encountered. In contrast to ConceptNet, which supplies mostly static knowledge about common words and terms, we seek to instill causal and sequential types of commonsense with the retraining using Atomic.

**HellaSwag** The HellaSwag dataset, an expansion of the SWAG dataset, was created to test commonsense reasoning in natural language inference (NLI) tasks. It consists of 70,000 textual descriptions designed to assess the ability to predict plausible outcomes based on a given scenario. For example, from an event such as a man being pulled on a water ski, HellaSwag might present multiple-choice outcomes where one needs to select the most logical next action based on common sense.

This dataset leverages Adversarial Filtering (AF), a technique where multiple discriminators are employed to refine and select the most challenging incorrect answers generated by a strong generative model. This approach makes the dataset simple for humans to solve (with an accuracy rate of 95.6%) but difficult for machines. Even when models are trained with extensive examples and tested on data from the same distribution, they struggle against this approach to dataset creation.

Unlike traditional datasets that may accidentally teach models to exploit specific, dataset-inherent biases and token statistics, the adversarial filtering and other steps taken during dataset creation largely prevent this. Further, HellaSwag includes diverse sources to broaden the context and challenge models with more complex scenarios.

The adversarial nature and the aforementioned features make this dataset and ideal candidate to test whether KIMERA is able to successfully

instill commonsense knowledge or not.

**Results**

We perform KIMERA re-training on both knowledge graphs jointly, and use BERT-base-uncased as the base of our experiments. The results of this experiment are depicted in Table 4.3. Unfortunately KIMERA did not have any significant performance impact on BERT-base-uncased performance in this dataset. There is a number of possible different reasons for this result.

**Table 4.3.** Overall Accuracy(%) across the different HellaSwag settings. BERT-base-uncased baseline result from HellaSwag Leaderboard[2]. No significant performance benefit of KIMERA is observed.

| Model | Overall Accuracy |
|---|---|
| BERT-base-uncased | 40.5 |
| KIMERA no-mask | 39.7 |
| KIMERA hard-mask | 40.7 |
| KIMERA soft-mask | 40.2 |

One primary concern is the potential mismatch between the content of the knowledge graphs utilized, and the requirements of the downstream tasks. KIMERA is based on the assumption that enriching transformer networks with domain-specific knowledge graphs can effectively augment their reasoning capabilities. However, if the knowledge graphs are not aligned closely with the needs of the commonsense scenarios presented in HellaSwag, the additional information may not be applicable or useful at all. In a similar vein, the knowledge included in those knowledge graphs might simply be too close to the information already implicitly contained in BERT-base due to its pre-training.

Additionally, the nature of commonsense reasoning itself is challenging. Commonsense involves implicit causal and relational understanding that may not be readily learnable through structured knowledge and the method of knowledge graph generation alone. Other types of data and likely other learning approaches might be required here.

[2]https://rowanzellers.com/hellaswag/

4. Efficiently Integrating Structured Knowledge Into Generic Transformer Models

Another aspect one might consider is the architectural limitations of the transformer model used in KIMERA. The adversarial filtering used in HellaSwag makes it specifically designed to challenge the capabilities of models like BERT-base after all. Additionally, the very simple and linear architecture of transformer models might simply not be able to reason in the required way, necessitating different inductive biases or architecture entirely. However, this is easily countered by the fact that RoBERTa does manage to achieve impressive performance on HellaSwag. While it is based on the larger `BERT-Large-uncased`, it does manage to handily beat the performance of both `BERT-base` and `BERT-large`[3].

In summary, the most likely culprit in the lack of performance improvements on this commonsense benchmark is the simple limitation of KIMERA that it requires the right alignment of retraining data, model, and downstream task. It is entirely possible that KIMERA could achieve much higher performance over `BERT-base` using different knowledge graphs, but this is outside the scope for this thesis.

## 4.6 Discussion and Analysis

In order to gain further understanding of how KIMERA changes the model's behaviour we perform an analysis of the attention heads before and after applying our method. In particular, we evaluate the hard-mask strategy in the CAPR setting, since the most significant improvements were achieved in this setting. This analysis further addresses and helps to confirm our hypothesis regarding the first part of this research question, whether generic transformer models are over parameterized.

**Model Over Parametrization** We showcase a 2D visualization of the hard mask determined by the iterative process outlined in section 4.3 in Figure 4.2 **A**. Here, yellow represents the important heads and purple the unimportant heads. Notably, 70.8% of attention heads show little effect on the performance in this instance. This high level of redundancy is compatible with our hypothesis and the performance gains we see for this set of tasks after applying KIMERA.

**Importance Adjustment**. KIMERA leads to an overall more homogenous distribution of Importance values as shown by Figure 4.2 **B** and **C**,

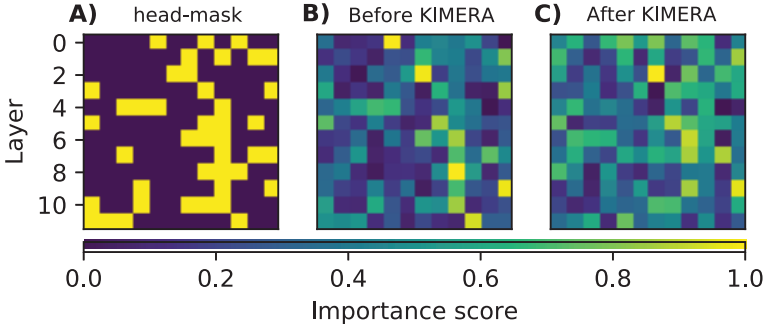



**Figure 4.2.** Analysis of $I_h$ and over parametrization before and after using KIMERA in clinical answer passage retrieval. **A)** Attention Mask generated in KIMERA Step I. **B)** Attention map $I_h$ of BERT-base before applying KIMERA. **B)** Attention map $I_h$ after applying KIMERA to BERT-base.

**Table 4.4.** Quantitative evaluation of $I_h$. It leads to a significant increase in $I_h$ for previously unimportant heads and leads to a slight decrease of previously important attention heads.

| Heads | $I_h$ Before KIMERA | $I_h$ After KIMERA |
|---|---|---|
| Frozen | 0.60 | 0.53 |
| Retrained | 0.17 | 0.37 |

which depict Importance Values $I_h$ before and after applying KIMERA respectively. While there are still a few attention heads that stand out regarding their Importance, there are much fewer unimportant (dark blue) heads.

Specifically we can demonstrate a significant and consistent increase in importance for the attention heads retrained with KIMERA as shown by Figure 4.3 **B** and Table 4.4. Figure 4.3 **B** shows that across every layer an increase in importance was reached for these heads. Table 4.4 confirms that the mean importance across the network in fact more than doubled, going from 0.17 to 0.37. These are strong indicators for KIMERA working

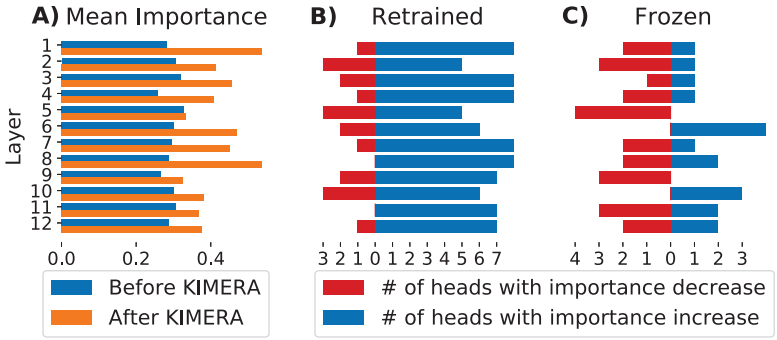



**Figure 4.3.** Statistics of shift in $I_h$ after applying KIMERA **A)** Shows a consistent increase of mean importance per layer. **B)** and **C)** show the effect of KIMERA split by retrained and frozen heads respectively. While retrained heads become significantly more important, the effect on frozen heads much less clear.

exactly as expected, utilizing the unimportant heads to store information gained from the knowledge graph that is then useful in downstream task.

For the frozen, previously already important heads, there is little change in $I_h$. Figure 4.3 **B** shows that across all layers a significant subset of the frozen heads become more important, and some less, there is no clear shift. Table 4.4 shows a moderate decrease in performance by only 12%. This means that the improvements gained in the retrained heads do not come at the expense of the capabilities the model already held. We therefore effectively combat catastrophic forgetting with the targeted retraining of unimportant heads.

In summary, this analysis highlights and confirms the existing over parametrization of transformer models necessary to answer research question 2. Further, it demonstrates that KIMERA not only leads to an improvement of the model in regard to performance, it does so by utilizing the previously superfluous parameters, leading to now less over parametrization, and a more efficient use of the parameters.

## 4.7 Limitations

The effectiveness of our proposed methodology in domain transfer is inversely proportional to how well the underlying multi-headed transformer model already does on a benchmark. This is evident in the stark contrast between the gains achieved by KIMERA in the CAPR and COP tasks as well the contrast between applying KIMERA on top of a generic BERT model, and applying it on top of BioBERT. The main factors behind this are the level of redundancy of the model for the task, which we gauge by the head-masks, and how complementary the target Knowledge-Graph is. Therefore, in order to apply this method, some preparatory work is needed. Users of KIMERA have to identify or create a knowledge-graph that contains key information, language, or technical words that both the model lacks, and which are required or helpful in solving the target downstream task, in order to make this an efficient and effective training method.

## 4.8 Summary

In this chapter we answered the question "Can over parameterized models be improved through Knowledge Graph Completion Retraining?". First, we used model compression techniques to gather insights on how over parameterized common transformer models are in the medical setting. In particular, we focused on Information Retrieval and Classification tasks.

Then we proposed a novel training methodology for improving pretrained Language Models and adapting them to the clinical domain. With that, we demonstrated the efficacy of utilizing structured knowledge from clinical knowledge graphs in a domain adaptation training scenario via knowledge graph generation. We showed that it leads to significant improvements in our models. We explored different strategies for freezing attention heads during retraining and achieve a significant and consistent improvement over strong baseline models. Our careful experiments confirmed our hypothesis that KIMERA adequately compensates for limited training data and domain knowledge. It makes large transformer models adaptable with limited effort and our results show that KIMERA manages

to improve on the already strong biomedical baseline of BioBERT.

We have showcased here how structured data in the form of a knowledge graph can be used for efficient transfer learning. While this opens up a new type of data in limited data niche domains and presents a very efficient approach both on both the data and computation complexity axes, it still necessitates having a substantially large knowledge graph. The next chapter will detail an approach where even that restriction is loosened. We will create a reinforcement learning environment based on a knowledge graph many magnitudes smaller than the knowledge graphs discussed in this chapter, and with it train a model with almost no underlying supervised data available.

Chapter 5

# A RL Environment for Differential Diagnosis and a Novel Learning Strategy to Solve It

## 5.1 Introduction

In this chapter we will discuss and analyze whether Reinforcement Learning can be a suitable alternative training strategy to the more established supervised learning, addressing research question 3. We do not aim here for a 1 to 1 comparison between the two learning approaches, as that would inevitably heavily favor one or the other depending on the downstream task, and they are simply too different to compare. Instead, we orient ourselves along the real world application of either type of approach, and model the problem in a realistic fashion, and then evaluate how reinforcement learning fares in this setting. The real world application we choose for this is DDx. We choose DDx because as a complex task in the clinical domain it highlights a few of the major challenges that Transformer models face in real-world applications. It is a task with a direct impact to human lives, necessitating a high degree of interpretability and accountability, and it is a task where only very limited data is available. The second challenge in particular is where Reinforcement Learning should shine.

In the DDx task doctors detect the disease(s) afflicting a patient by applying a series of examinations, with each one uncovering new symptoms of the patient's underlying disease and thereby narrowing the set of possible diseases. This task is challenging because of the vast number of diseases that exist with very similar sets of symptoms. Compounding difficulties are time, costs, and risks involved with certain examinations and

treatments. For example, examinations such as CT-scans involve unwanted exposure to radiation doses, and invasive exploratory procedures such as laparoscopy carry the risk of complications. Furthermore, the usage of powerful diagnostic tools such as an MRI might be cost-constrained and limited, thus other avenues of examination might have to be considered first. Depending on the disease, the patient's condition might also deteriorate while different examinations are applied. Therefore, a quick diagnosis is paramount.

Uniquely, we frame this task as a natural-language-based online-learning problem. Phrasing differential diagnosis in this fashion introduces a set reduction problem. From this point of view, the agent has the goal to reduce the set of possible diseases to the singleton set of only the patient's actual disease in the shortest amount of steps. In this particular application of set reduction, planning ahead is critical since both, the disease and largely the composition of the symptoms, are unknown, and there are complex interactions between symptoms and procedures. It is not enough to choose in each reduction step the examination that excludes the most diseases by detecting or not detecting certain symptoms. Instead, the whole trajectory has to be considered. Furthermore, Examinations have to be chosen in combination with treatments such that they complement each other leading to the largest disease set reduction overall while considering the deterioration of the patient's health. This long-term planning requirement favors Reinforcement Learning, as has been demonstrated in a number of different applications[KVC+21; SHS+17; SHM+16].

For our experiments we create a novel Reinforcement Learning Environment, where the agent simulates a doctor. In each episode, it has the task of diagnosing one patient. We go beyond this narrow definition of DDx and in addition to diagnosing, we task the agent with treating the disease. Furthermore, the agent should learn to remedy symptoms that might be particularly severe and harmful for the patient, because these symptoms might lead the patient's condition to deteriorate before the proper diagnosis and treatment can be found.

**Medical Text and Transformers for RL** We focus specifically on creating a text-based environment for this task, since most of the information necessary for doing differential diagnosis is naturally text-based in the

form of the patient's history or doctor's notes. The choice of transformer language models as a baseline to solve this environment is natural.

However, Transformers have been shown to struggle with instability in online reinforcement learning problems when acting as a policy[PSR+20]. To address this issue, we propose a novel training approach based on an auxiliary masked language modelling objective that complements the reinforcement learning training. Using this approach, we outperform other baselines including a standard transformer, but we show that our environment remains challenging in this online setting.

**Analysis with Medical Professionals** We create our OpenAI-Gym based environment with data from online medical resources curated by medical professionals. We further evaluate this data, as well as the resulting environment, with medical professionals. Lastly, we perform an in-depth quantitative and qualitative analysis on trajectories of our best-performing agent to highlight strengths and shortcomings of the learned policy. In this analysis, we also show that symptom-examination overlap among diseases is a major factor contributing to the difficulty of this task.

To summarize, the contributions discussed in this chapter and published as [WFL+23] are the following:

1. To the best of our knowledge, we are the first to phrase the full differential diagnosis problem as a text-based reinforcement learning scenario in an online setting. We further expand this problem with the treatment of the patient.

2. We create an environment and release it together with the underlying data that we label with the help of medical professionals[1].

3. We propose a masked language model (MLM) objective as an additional loss to improve online reinforcement learning with transformers, environment modelling and regularization, and show that this approach outperforms other baselines on this task.

[1]For the sake of anonymity, source code and data will be added on publication

5. A RL Environment for Differential Diagnosis and a Novel Learning Strategy to Solve It

4. We provide an in-depth qualitative analysis on the trajectories of our best model.

The remainder of this chapter is structured as follows: in Section 5.2 we discuss related, previous research, in Section 5.3 we detail both the DDxGym environment, and the knowledge graph it is based on, and in Section 5.4 we describe our model and training approach. Then, in Section 5.5 we detail our experiments, in Section 5.6 we discuss our experimental results, as well as our findings from the trajectories of our best agent, and we close with Section 5.10 summarizing this chapter.

## 5.2 Related Work

In this section we discuss the specific research related to our transfer learning via reinforcement learning approach, building upon the foundation of related work laid in Section 2. In particular, we discuss the topics of Reinforcement for medical diagnosis, medical knowledge graphs, and reinforcement learning using transformers.

### 5.2.1 RL in Automated Diagnosis Systems

Given the interactive nature of the problem of the clinical diagnostic process, RL has been used as a suitable framework. Tang et al. proposed an ensemble of neural networks corresponding to anatomical parts of the body which questions the patient for symptoms and diagnoses diseases [TKC+16]. They follow up on this work by introducing hierarchical reinforcement learning (HRL), contextual demographics, as well as hereditary and medical history information [KTC18]. Similarly, Yuan et al. decomposed the diagnostic process by aligning an RL agent trained to uncover symptoms with a supervised classification objective for the diagnosis step [YY21]. Furthermore, an automatic symptom detection system based on a graph-memory-network agent was proposed [LLG20].

[WLP+18] at first glance is closely related to our work. Authors create a dialog system where an agent learns a policy to communicate with the patient and ask questions about their state. This work has a very similar goal and outcome to ours, but focuses entirely on the conversational aspect,

and yet bases the environment on discrete information vectors instead of natural language.

In contrast to all of these works, we define the environment to produce only natural language observations and approach the training of the agents purely with NLP methods and models. Further, we add the additional process of treatment to the patient episodes, and we don't make a distinction between the action spaces of examinations and treatments.

### 5.2.2 Structured Medical Knowledge

Several authors have focused on creating language definitions to express the medical, structured knowledge in computer interpretable guidelines CIGs for clinical decision support systems (CDSSs) [SAD+01; DHB+01; BPT+04; FJR98]. In our work, we didn't focus on the formalism surrounding medical knowledge representation, rather we created a simple knowledge base to simulate interactive paths via an RL environment. An agent trained in this environment then proposes examinations and treatments analogous to a CDSS.

Curated knowledge bases are at the core of many commercial CDSSs [HRK+20; NKS+19; RBP+18]. More generally, approaches such as UMLS [LHM93] or SNOMED [RC96] aim to unify several biomedical concepts into abstract general-purpose knowledge graphs. Being accordingly general, these knowledge graphs are not specific enough with respect to the symptom-procedure relations required by DDxGym. In our work, we differ by letting an RL environment be defined by a simple, yet extensible, knowledge base that encompasses multiple diseases with very concrete edges and semantic descriptions which we make openly available.

### 5.2.3 Reinforcement Learning using Transformers

Given their wide success in sequence processing tasks in supervised settings, transformers are increasingly used in combination with reinforcement learning. Transformers modelling policies face challenges such as learning stability and low sample efficiency [LLL+23]. Parisotto et al. highlight the problem of stability and tackle memorization tasks with a gating architecture in place of the transformer residual connections [PSR+20]. In

contrast, our approach doesn't modify the transformer architecture but rather adds an additional concurrent objective to stabilize the learning of a policy.

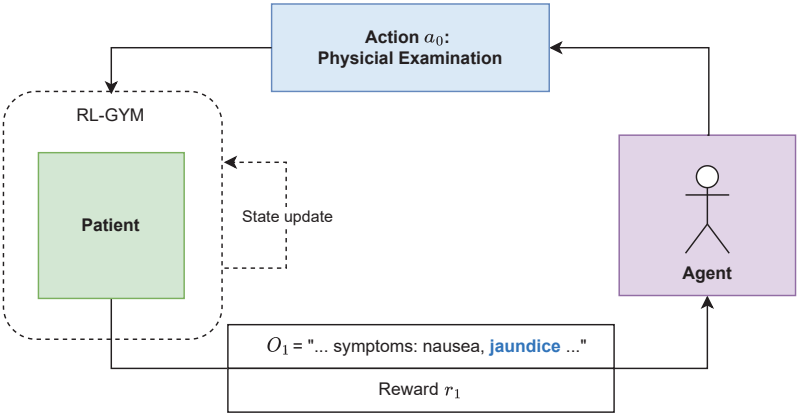
Autoregressive transformer-based language models can be used as appropriate policy initializers for learning in environments adapted to yield textual observations of the state [LPP+22]. Yao et al. use language models which are fine-tuned on human gameplay to filter for admissible actions to serve as input to a policy [YRH+20]. These works don't use the language models as a policy but only leverage the common-sense grounding they acquired during pre-training. Instead, our approach utilizes a transformer to model the policy. A parallel avenue to solving decision problems is translating them to a supervised setting. Chen et al. rephrase RL as a supervised learning problem solved by an auto-regressive model of reward, states, and actions [CLR+21]. Similarly, Carroll et al. model reward, state, and actions with bidirectional transformer encoders and masked language modelling [CPL+22]. We differ from these approaches since we don't fine-tune the LMs in a supervised setting but rather we choose the transformers to explicitly model the policies that we train exclusively in an online setting to tackle a purely textual environment.

More recently and to the wide attention of the public, in systems such as InstructGPT [OWJ+22b] and ChatGPT [SZK+22], large language models based on transformers have incorporated RL mechanisms to yield impressive results in interactive settings. Our approach strictly differs from these works, since we train our models in an explicit online RL environment, we don't utilize a reward approximation model, and our agents are not trained in a supervised setting.

## 5.3 DDxGym Environment

The goal of this environment is to model the challenging task of differential diagnosis in a realistic manner with sensible simplifications.

We understand the problem of diagnosing, and subsequently curing a patient, as a partially observable Markov decision process (POMDP) described by $(S, A, P_a, R_a)$. Figure 5.1a shows an overview of the DDxGym environment. An example episode trajectory can be seen in 5.2.




**(a)** Overview of DDxGym and the reinforcement learning setup. Given an initial observation that includes a symptom ´nausea´, the agent chooses $a_0$ (physical examination). This results in an additional symptom ´jaundice´ being discovered, as seen in observation $O_1$ with the corresponding reward $r_1$.

### 5.3.1 Environment Definition

**States and Observations** Each state $s \in S$ consists of:

1. The patient's disease $d \in D$
2. A set of symptoms $z_x$ with their respective states (hidden, discovered, cured)
3. The history of procedures the agent has already applied to the patient
4. A decaying integer value that denotes the "health" of the patient $H$

The observation $O$ given to the agent is a text sequence that describes the patient's *discovered* symptoms, the applied procedures, and which symptoms have already been cured, emulating an electronic health record (EHR). The hidden entities in the state, such as the undiscovered symptoms, the value of $H$, and the disease $d$ are not observable. This makes the environment as a whole partially observable. An example of such an observation can be seen in Figure 5.1a.

```
----Initial State------------------------------
List of symptoms   : nausea

----step 1 [physical examination]--------------
List of symptoms   : nausea, jaundice, fatigue
Treated symptoms   :
Applied procedures : physical examination

----step 2 [run test - bloodtest]--------------
List of symptoms   : nausea, jaundice, fatigue
Treated symptoms   :
Applied procedures : physical examination,
                     run test - bloodtest

----step 3 [use device - ct]--------------------
List of symptoms   : nausea, jaundice, fatigue,
                     liver cancer discovered
Treated symptoms   :
Applied procedures : physical examination,
                     run test - bloodtest,
                     use device - ct

----step 4-[procedure - radiotherapy]-----------
List of symptoms   : ...
Treated symptoms   : liver cancer discovered
Applied procedures : ...
```

**Figure 5.2.** A history of observations for a full example episode of our best agent interacting with DDxGym treating liver cancer.

**Actions** The action space encompasses all the procedures that are available to the agent. There is no distinction in this space among the actions of examinations or treatments. The actual diagnosis of the disease occurs naturally with the examinations. This is in contrast to other works [YY21], which model the disease prediction separately. Our definition of the action space doesn't induce any structure of the problem for the agent which makes it more challenging.

**Episode Dynamics** Each episode begins with a patient with one randomly sampled disease. The value of $H$ is initialized to a positive integer. The exact value of this initialization is a hyperparameter and one of the main factors that determines the budget of interactions that the agent can have with the patient. The observation in step 0 includes one symptom

and no procedures. The symptom is sampled by occurrence probabilities from the set of all symptoms which are not the main symptom and have an initial onset. This emulates the chief complaint i.e. the health issue which was the reason for the patient to visit the doctor.

Each step in the environment constitutes choosing exactly one procedure to apply to the patient. With every action, symptoms may be detected or treated, and the observations and reward are updated accordingly. Further, with each step, regardless of which procedure the agent applies, the patient deteriorates, which reduces the value $H$. How quickly the value deteriorates depends on the severity of the disease, and the severity of the untreated symptoms of the patient in $z_x$. This incentivizes the agent to learn a policy that treats very severe symptoms (e.g., internal bleeding), before focusing on diagnosing the underlying disease. Each episode terminates, if either the value of $H$ becomes negative, or if the disease is detected and treated.

**Reward** Our environment features a mostly sparse reward structure. The largest positive rewards are given only when the disease is diagnosed and subsequently treated. Smaller positive rewards are given for uncovering and treating symptoms that are not the main symptom. The value of this reward depends on whether the symptoms are diagnosed or treated, and on how severe they are. In each step where the chosen action does not lead to the detection or treatment of any new symptom, a negative reward is returned with the value of the deterioration of $H$ for this step. This steady negative reward encourages the training of policies that treat the patient in the smallest amount of steps possible, which is desirable.

To summarize, the reward structure of the DDxGym environment is

$$r_t = \begin{cases} 1000 & \text{if the main symptom was cured} \\ 100 & \text{if the main symptom was discovered} \\ (50, 20, 10) & \text{if a non-main symptom was cured} \\ (20, 10, 5) & \text{if a non-main symptom was discovered} \\ \sum z_{x_s} & \text{otherwise, where } z_{x_s} \in (-5, -2, -1) \end{cases} \tag{5.3.1}$$

with $r_t$ being the step-wise reward, and $\sum z_{x_s}$ being the sum of the severity of all untreated symptoms afflicting the patient. Values in parentheses are

for high, medium, and low severity symptoms respectively. This reward structure results in a lower bound of cumulative episode reward that is equal to $-H$, and an upper bound of 1200 for our data.

**Table 5.1.** DDxGym knowledge graph statistics. The resulting environment actions are defined by the number of examinations and treatments, amounting to a total of 330.

| Environment Concept | # distinct entities |
|---|---|
| Diseases | 111 |
| Symptoms | 384 |
| Examinations | 154 |
| Treatments | 176 |

### 5.3.2 DDxGym-Knowledge Graph

In order to create the DDxGym environment as described in the previous section, we need a structured data definition that captures the medical concepts and interactions of procedures, symptoms, and diseases, i.e., a knowledge graph. Finding this type of data is challenging. While there are a few proprietary knowledge graphs available that contain this type of information, they are not freely available for research ([NKS+19; HRK+20] or the work of Infermedica[2]). Widely used open medical knowledge graphs such as UMLS [LHM93] and SNOMED [RC96] don't capture the treatment-symptom, or examination-symptom relations that would be needed to specify the environment. For that reason, we decide to label our own data and create a suitable knowledge graph. We make this data publicly available, and we hope that in the future our knowledge graph will be expanded with even more diseases. We expect that publicly available data promotes the release of more exhaustive knowledge graphs in this direction which could be applied to our methods[3].

[2] https://developer.infermedica.com/docs/v3/medical-concepts

[3] We release our knowledge graph upon publishing

**Labelling process.** The basis of our knowledge graph is built from educational disease resources on medical pages[4] which are curated by medical professionals. A group of individuals with experience in biomedical NLP extracted under strict guidance (a) *diseases*, (b) *symptoms*, (c) *examinations*, and (d) *treatments*, and the relations between them. To make the environment more realistic, we also labelled semantic descriptions for the symptom entities and their relations:

1. a probability for each symptom to occur
2. the main symptom characterizing the disease
3. a time horizon on when the symptom might develop
4. a severity that determines how quickly the patient deteriorates

Figure 5.3 illustrates how the labelled entities relate to each other in our knowledge graph. In a second round of labelling we evaluate this data by showing disease nodes of this knowledge graph to medical doctors who fine-tune the relations as well as add or replace symptoms and procedures. Finally, the entities of the knowledge graph are normalized, acronyms are disambiguated and duplicates are merged. The entities, relations, and semantic descriptions related to one example disease can be seen in Table 5.2.

**Knowledge Graph Statistics** Table 5.1 shows the results of this labelling process. After clean-up and quality assessment, the environment is composed of 111 diseases that are diagnosed and treated with a sum total of 330 unique procedures. This represents a relatively large action space and one of the main factors that make this environment challenging. This complexity grows as more diseases and procedures are added, as is the case of highly curated commercial knowledge bases. While we are limited in this work to our knowledge graph, our approach is generally applicable to these much larger commercial alternatives.

[4] `www.nhsinform.scot`, `www.mayoclinic.org`, `www.nhs.uk`

5. A RL Environment for Differential Diagnosis and a Novel Learning Strategy to Solve It

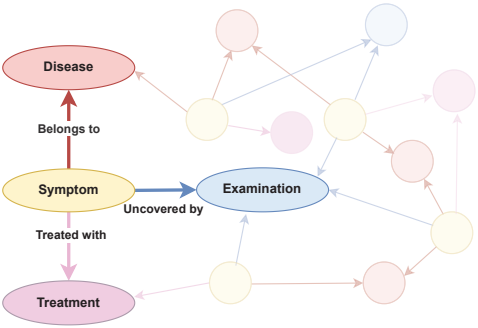



**Figure 5.3.** Disease relations in the DDxGym knowledge graph. The same procedures might connect to different symptoms and therefore multiple diseases.

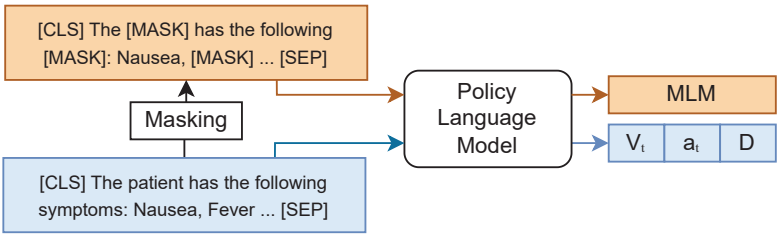



**Figure 5.4.** Model Architecture. For each environment step there are two forward passes over the same model. First, the observation $o_t$ is used to predict the value of the current state $V_t$, choosing the agent's next action $a_t$, as well as predicting the patient's disease for the T+PD baselines. In the second pass, the observation is masked and then used to train the masked language modelling objective.

## 5.4 Methodology

Previously in Section5.3, we have detailed the POMDP that describes our DDxGym environment. This section will illustrate now how we use Reinforcement Learning to train an Agent to solve this environment.

**Table 5.2.** *Acute pancreatitis* in our knowledge graph. There are four different symptoms. The disease identifier is the main symptom. Each of the symptoms has at least one *examination* that uncovers it. Not all of the non-main symptoms might actually affect the patient, this is governed by the *probability* field. They also might only appear after a few environment steps which is determined by the *onset*. While the goal is to treat the actual pancreatitis, the agent might find it useful to treat the *fever* and *nausea* if they are present, since their *severity* leads the patient to deteriorate more quickly. The symptom *jaundice* however, can not be treated directly as there is no *treatment* for it in our data.

| Symptom | Examination | Treatment | Severity | Onset | Probability | Is main? |
|---|---|---|---|---|---|---|
| Acute pancreatitis | Run blood lipase and amylase test | IV (Fluids) | high | initial | always | yes |
| Fever | Physical Examination - Body temperature | antipyretics | mid | short | medium | no |
| Nausea | Interview - nausea | antiemetics | mid | short | medium | no |
| Jaundice | Interview - visual | | low | short | medium | no |

### 5.4.1 Algorithm

We choose IMPALA[ESM+18] as a Reinforcement Learning algorithm. While other choices are sensible and our environment is not algorithm specific, IMPALA exhibits a number of features that make it suitable for our use case. Most notably it is one of very few RL Algorithms that is highly parallelizable, enabling it to sample large amounts of episodes from our environment, which has low computational cost. This gives it a potential advantage over more sample efficient algorithms such as PPO [SWD+17] which are more suited to slow and costly environments.

IMPALA is an actor-critic algorithm and requires a model that, given an observation, outputs in each step both a distribution over the action

space $a_s$, and an estimation of the Value function for the current state $V_s$. To encode these observations we choose a transformer language model as detailed in the following section. Utilizing a model in this way it functions both as a policy, and as a Value Function estimator. As discussed previously, research[PSR+20; LLL+23] has shown that transformers are unsuitable for this purpose and result in highly unstable learning and polices. Nonetheless, they are the most powerful sequence models to date, therefore we develop a novel strategy to address this problem. Specifically, we introduce additional learning objectives that stabilize learning. The use of the transformer model and the additional objectives are detailed in the following.

### 5.4.2 Transformer Encoder

The observations supplied by the DDxGym environment are already in the form of tokenized text. We therefore apply a pre-trained transformer language model in a straightforward way to produce a contextualized representation. Specifically we follow the common approach of pooling this representation by taking the vector of the [CLS] token as a sequence representation. We discuss the choice of pre-trained transformer in Section 5.5. In order to generate $a_s$ and $V_s$ we add two single layer feedforward networks on top of the pooled representation, a softmax for $a_s$ and a linear activated regression for $V_s$.

### 5.4.3 Additional Learning Objectives

We evaluate two additional objectives to assist the transformer model in learning the policy, Masked Language Modelling, and Disease Prediction. These objectives run ***concurrently*** with the reward-based IMPALA objective. To that end we compute the loss as a weighted sum of IMPALA loss and our additional losses. Figure 5.4 demonstrates this parallel learning of different objectives.

### Masked Language Modelling Objective

In order to ground the representations in the language of the observations we adopt masked language modelling on the observations as an objective. Our hypothesis is that this supports the model in understanding the observations, and it shifts the focus from the purely control-based aspect of generating action distributions, to generating good representation of the observations.

We follow the training regime of [DCL+19] and in each step randomly mask 15% of the tokens. An additional feedforward-based output head generates the predicted tokens based on the pooled representation. The aggregated, token-wise cross entropy of the masked language modelling forms the loss of this objective. When applying this objective, we have to run 2 forward passes in each environment step. One with the input of the masked observations for the masked language modelling objective, the second with unmasked observations for the other objectives. Since forward passes are generally very fast, this has little impact on performance, however. In the backward pass losses can be combined so one backward pass suffices.

This learning of environment representations goes in the direction of model-based reinforcement learning. We only predict (parts of) current observations however, and not future observations. We therefore do not learn transition dynamics. Exploring that in concert with transformer models we leave for future research.

### Disease Prediction Objective

The second objective is a simple supervised classification over the set of diseases available in the environment. In each environment step the model is tasked with predicting the disease of the patient. Mainly this objective was chosen as another way to ground the observations, and link them more closely with the underlying state of the environment. However, this objective has implications for explainability of the resulting agent as well. When applying this agent in the future an examination or treatment it chose could be explained and legitimized by which disease it predicted the patient to be afflicted by. This would further help a medical professional in

deciding a course of action and increase trust in the model. We implement this classification objective with a single linear layer output head on top of the pooled representation using the [CLS] token. It runs in parallel to the other objectives in each step, and unlike the masked language modelling objective can be calculated in the same forward pass as the reinforcement learning outputs.

## 5.5 Experimental Setup

We conduct multiple experiments on our DDxGym environment using the Agent described in the previous sections. The transformer is initialized from the pre-trained checkpoint and the additional output heads are randomly initialized. The reinforcement learning training is then run until a maximum number of total environment steps is reached. To ensure optimal training of our agent we further perform a hyperparameter search. In order to get the most of that search with limited hardware we focus only on optimizing the learning rate, the proportion of samples drawn from the replay buffer, and which exploration strategy the agent follows. The exact parameters can be seen in Table 5.3.

**Table 5.3.** Parameters for training transformer baselines.

| hyperparameter | Value |
|---|---|
| learning rate | $[3 \cdot 10^{-7}; 3 \cdot 10^{-5}]$ |
| replay proportion | [0.1;0.5] |
| replay buffer size | 128 |
| train batch size | 800 |
| inference batch size | 50 |
| exploration strategy | [$\epsilon$-greedy, stochastic sampling] |
| sequence length | 128 |
| maximum training steps | 80M |

For the purpose of evaluating our agent we focus on the mean episode reward $\bar{r}$ and mean episode length $\bar{l}$. With how the DDxGym environment is set up, these should be inversely proportional to each other, and an

ideal agent should achieve rewards close to the theoretical maximum of 1200, and low episode lengths close to the theoretical minimum of 2. These results would describe an agent that efficiently solves patients with a large variety of diseases.

## Models and Baselines

In order to get a holistic view of both how challenging our environment is, and how well the transformer policy functions with our additional objectives, we evaluate multiple different Agents. We detail those Agents in this section:

**T** Stands for the basic transformer baseline. We use the pre-trained transformer as an encoder, and do not use our additional objectives.

**T+PD** is the transformer model with the added disease prediction objective, and its corresponding output head.

**T+MLM** is the transformer model with the added masked language modelling objective, and that corresponding output head.

**T+PD+MLM** is a model combining both additional objectives, learning both, and the IMPALA objective, with equal weights.

**Text-sequence LSTM** With the aim of demonstrating that our additional objectives manage to address the transformer policy's challenges and thereby elevate it over the more common LSTM policy, we evaluate an LSTM[HS97] baseline. This gives a direct comparison between the two architectures. In particular, we choose a 3-layer bidirectional LSTM with dropout and a hidden layer size of 256. It shares the vocabulary and word embedding with the transformer model, and thereby a small amount of its pretraining.

**Random** is a fixed policy choosing actions at random. This gives us a lower bound for performance.

## Choosing a Transformer Model

When selecting pre-trained transformers as encoders, there's a wide array of pre-trained BERT and similar architectures available. In reinforcement learning, smaller and quicker models are often preferred because RL requires extensive sampling for effective training. Yet, recent trends in supervised learning have shown that, especially for transformer models, larger variants tend to outperform their smaller counterparts. Thus, to select a model for further experiments we tested three different transformer encoders: BERT-base-uncased, ClinicalBERT (a version of BERT pre-trained on MIMICIII patient admission notes), and a more compact version of BERT-base-uncased, BERT-small. BERT-base-uncased is still one of the most popular generic transformer models, ClinicalBERT we chose since with its clinical pre-training it should have a big advantage over generic models, and BERT-small as a model that is small and much faster to train than the previous 2.

The outcomes of these tests, depicted in Figure 5.5, show all three models achieving a mean reward ranging from 100 to 200. Given that the optimal mean reward is around 1200, it's clear that the performance level of these models is relatively low. ClinicalBERT edges out slightly ahead of the other two, likely owing to its domain-specific pre-training. Both ClinicalBERT and BERT-base-uncased experienced noticeable performance dips however, reaching a minimum reward of -200, underscoring their instability. In contrast, BERT-small, while performing on par with BERT-base-uncased, demonstrated greater training stability.

This stability in BERT-small could be attributed to the larger batch sizes it accommodates and its overall fewer parameters, which likely contribute to normalization effects. Additionally, BERT-small enabled us to execute more than 1.5 times more training steps than the larger models within the same timeframe.

Given the limited resources, the apparent stability advantages, and only minor performance discrepancies between the models overall, we opted for BERT-small in our further experiments.

**Implementation** We share here a few key details regarding the implementation of the environment and the agent. The environment is based

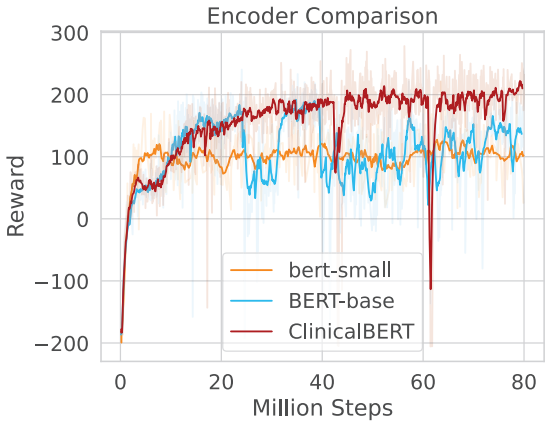



**Figure 5.5.** Evaluation of various pre-trained transformer language models as policies in the DDxGym environment. Performance of the three models is comparable, while BERT-small operates at a substantially higher speed. To improve readability, exponential moving averages are presented with $\alpha = 0.85$.

on OpenAI's Gym[BCP+16] Framework. This is by far the most popular framework for Reinforcement Learning environments, and it makes our environment highly portable to different algorithms and other frameworks. For implementation of the training and evaluation code we make prominent use of the Ray and RLLib[LLN+18] frameworks. The IMPALA algorithm we use stems also from RLLib. For the transformer architecture, as well as the loading of the pre-trained checkpoints we use the `huggingface` [WDS+20] library.

We run the previously described experiments on a DGX100, using 6 A100 GPUs. Since inference and sampling from the environment are much faster than calculating the gradients in the backward pass, we use 4 of these GPUs as learner GPUs in IMPALAS learner-actor architecture, and 8 actor workers which share the other 2 GPUs via GPU sharding. In our experiments this gave us the best performance for our hardware.

## 5.6 Experiments and Results

In this section we will detail the different experiments we performed on the DDxGym environment and their results. We start with and experiment training an agent on the DDxGym with a toy knowledge graph dataset and then discuss our main, published results with the real data we collected as previously described. Finally, we detail two additional experiments, which are ariming to improve our model using *action embeddings*, and replacing the transformer model with an entirely different architecture respectively.

### 5.6.1 Initial Experiment: Project Hospital Data

Since data collection is a costly and time-consuming process, and there are no openly available datasets available that matched our criteria for this research, we built a proof of concept on an artificial dataset. We discuss here briefly this artificial dataset, and then the results we reached on this dataset that lead us to pursue this research further.

The experimental set-up is identical to our experiments with the real data. We use IMPALA as our Reinforcement Learning algorithm, and use DDxGym as described in previous sections. The only difference is the knowledge graph feeding the gym, which leads to different diseases, patient trajectories, and overall complexity.

#### Dataset

The source of this dataset is the video game "Project Hospital"[5]. In this game the objective is to build and run a hospital as its director. To that end one has to employ and train doctors and nurses and treats patients. The foundation of this game is a complex engine that governs the diseases which patients visit the hospital with, and how the doctors interact with them.

This engine is built on top of a knowledge graph representing exactly the kind of knowledge we seek for our environment, concerning diseases, their symptoms, and the examinations and treatments that relate to them. In agreement with the developers we extract the knowledge graph from

[5]https://oxymoron.games/projecthospital/

the game files and process it to fit our needs. Of course, this data is not entirely realistic. It is narrow in scope, and the interactions between different entities in the knowledge graph are simplified as concessions to it being a video game. Nevertheless, it gives us a good starting point for our research, since at least on a surface level it is medically accurate.

**Results**

We conduct our experiments using a basic `BERT-base-uncased` as our policy model, without our improved training approach. We train for 40 million steps and perform a very rudimentary hyperparameter optimization, optimizing inference worker and training worker batch sizes for computation speed, as well as the learning rate. In order to get a better understanding of how challenging this task is, we perform experiments with different numbers of diseases. We shuffle the set of 125 total diseases with a fixed random seed for repeatable experiments, and then take the first $3, 10, 25, 50$ diseases as overlapping subsets and limit the environment to produce patients with only those diseases. We further limit the action space to only examinations and treatments that are relevant to each subset of diseases. This has a significant impact on the initial learning process.

We report the Mean Episode Reward metric, as we will do with the real data. Only the version of the environment with 3 diseases is solved without difficulty by the transformer agent, reaching close to maximum mean rewards in a few hundred thousand environment steps. Even 10 diseases pose already a significant challenge to the agent, learning is much slower, and even after 40 million steps only a reward of $\tilde{7}50$ was reached. There is little difference between the experiments of 25 and 50 diseases. Both of them get stuck around the 200250 reward mark. This is a symptom of the agent being proficient at consistently diagnosing the disease of the patient, but then failing to treat it. With the agent failing to achieve the high rewards from treating patients, it more and more chases the easier diagnostic rewards and thereby gets stuck and the probability of it choosing treatment actions versus examination actions becomes slimmer and slimmer. For the full dataset of 125 diseases the agent falls short of even diagnosing the disease in a significant number of cases, leading to lower and often even negative rewards.

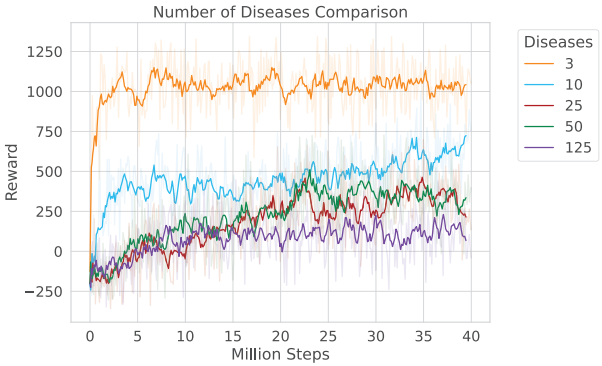



**Figure 5.6.** Results in the DDXGym Environment with Project Hospital data, limiting the environment to different subsets of diseases. While the smallest set is very easily solved, even just 10 diseases lead to a considerable challenge for the transformer policy.

We conclude that the problem of solving differential diagnosis in an interactive reinforcement learning setting is indeed challenging. Our interpretation in particular, which tries to mimic as closely as possible the process as it is applied in hospitals, by following full patient trajectories with requisite intermediary examination steps and subsequent treatment steps, makes this task complex. A basic transformer is not able to solve this task even when limiting the scope to a small set of diseases and so additional techniques are necessary to improve stability and learning of the transformer.

### 5.6.2 Quantitative Results on DDxGym

Figure 5.7 shows the results of evaluating the experiments previously described. It shows that all of our baselines beat the random policy by a wide margin in the mean episode reward metric. In contrast, some of the compared models, $T$ in particular, barely achieve shorter episode lengths. In that extreme case it is the result of the model learning to diagnose and treat symptoms, but never learning to treat the main symptom of the

disease. We note that *T+MLM* significantly outperforms other baselines in both mean episode reward and mean episode length. We also observe this model's reward is significantly more stable than *T+PD*, *T+MLM+PD*, and the *LSTM* models. Additionally, the mean episode length is the most stable in comparison to the other models. While the *LSTM* outperforms BERT-small without any auxiliary objectives, it falls short of *T+MLM*. This is an interesting result since transformers have superseded LSTMs in supervised learning, the RL setting keeps being challenging and DDxGym is no exception. The additional disease prediction objective in *T+PD* and *T+MLM+PD* leads to increased instability, and did not achieve improved performance. We believe that this objective distracts the agent from the actual control problem of diagnosing and treating. In summary, while there are still improvements left to be made, the quantitative results show that for a large amount of diseases reinforcement learning is able to very efficiently and correctly diagnose and treat patients in this challenging scenario. This gives very positive evidence to our research question.


**Figure 5.7.** Comparison of five different baselines on the DDxGym environment. The transformer model with auxiliary masked language modelling objective (*T+MLM*) clearly outperforms other baselines, both in mean reward (left) and in learning stability. This is also noticeable in episode length (right), showing it manages on average to treat patients in the shortest amount of steps.

### 5.6.3 Additional Experiment: Action Embeddings

The actions an agent in the DDxGym environment takes are complex, and their impact not always straight-forward. Many actions also relate

to each other. For example, `physical examination: arms` and `physical examination: legs` are very similar in nature, but target different regions of the body. These connections and relationships are lost when simply choosing actions from a distribution as is commonplace in RL. The only method the agent can then use to learn the meaning of these actions then is trial and error, guided by exploration strategies. This is a costly process requiring millions of training steps.

We hypothesize, that this process, and by extension the overall training of the agent, becomes easier and faster when the agent chooses actions in a more meaningful way. Our goal is to ground the actions in a similar fashion to grounded language learning[ABB+22]. This idea is not new. The seminal work in the field of action embeddings is [DES+15], where authors learn a policy based on pre-known action embeddings. To choose actions, the policy produces a *proto-action* in the form of a vector, which then selects the actual action by a k-nearest neighbours (kNN) search with $L_2$ distance between the *proto-action* and the action embeddings. We adopt a similar method, but choose cosine similarity as a distance metric and do not use kNN. Using cosine similarity is not efficient for large action spaces of up to a million actions as discussed in [DES+15], but is a natural choice for DDxGym with its limited number of complex actions, and text-based representations.

**Method**

In order to pre-compute our action embeddings we extract textual descriptions from UMLS[Bod04]. These descriptions are usually a single sentence and describe the procedure performed. For procedures that lack a UMLS description we manually write them. We then generate embeddings by processing these textual descriptions with the same model that will subsequently act as our agent. We choose to use the same transformer model for both, in order to ensure that the vector spaces of the embeddings and the generated proto-actions are similar. Specifically, we experiment with taking the `[CLS]` token vector or the mean of all token vectors, from the last hidden layer of the network as a representation. These are computed ahead of the reinforcement learning training.

During training and in each step, we compute the cosine similarity between all actions embeddings and the last hidden layer representation of our policy transformer. We follow an $\epsilon - greedy$ exploration strategy, and therefore either choose an action at random to aid exploration, or choose the action with the highest cosine similarity:

$$a = \arg\max \frac{E \cdot a_p}{\|E\| \|a_p\|} \tag{5.6.1}$$

such that $E$ is the Embedding matrix with all action embeddings, and $a_p$ is the proto-action representation generated by the agent. We experiment with both keeping action embeddings frozen during this process, and with further learning them during RL training.

**Discussion**

Unfortunately, none of our experiments with action embeddings lead to any noticeable improvement over the base transformer, or our approach with masked language modelling. At most, results were similar, and often times worse. We discuss here the reasons and possible avenues for future research on this specific experiment.

It is clear, that the additional detour over cosine similarity makes loss attribution more challenging for the model. Especially with frozen action embeddings, the task shifts from classifying the correct action for each state or observation, to producing a representation that points to a potentially arbitrary point in the vector space. In particular for a small model this can be challenging. Whether this approach works naturally rests largely on how representative the action embeddings are, and how well they are differentiated in the vector space. Our approach of generating BERT-style embeddings from a very short description is likely not good enough for this purpose.

We halted our experiments in this direction at this stage. We offer however a few ideas for possible improvements over our experiments. First, better textual descriptions could be sourced that extend to larger paragraphs and more detailed descriptions of the procedure. This could lead to a significantly more nuanced representation. Further, it might be worthwhile to experiment with different transformer models for the

generation of the embeddings. A medically trained transformer such as BioBERT[LYK+20] might lead to more useful representations, even if the vector space is entirely different from what the policy transformer produces, at least initially. Lastly, there are a number of strategies (e.g. [CTK+19; PML21; PMK24]) to continually improve action embeddings during reinforcement learning training. This seems like an important step in order to balance the grounding aspect of pre-computed action embeddings and the optimization of the reinforcement learning objective.

### 5.6.4 Additional Experiment: Fruitfly

We discuss in this section an alternative architecture to the transformer that we experimented with as a policy. The main goal of this thesis is to develop and discuss efficient approaches to domain-specific tasks such as differential diagnosis. While small transformer models such as the BERT-small used in this chapter are highly parallelizable and fast for their size, there are faster architectures out there. One such architecture is the Fruitfly[LRH+21].

Authors of this architecture take inspiration from the brains of fruitflies, and learn a simple and lightning fast word embedding model, which we exploit as our policy. As a comparison, the record of pre-training BERT-base within 47 minutes required 1472 Nvidia V100 GPUs, while the fruitfly can be pre-trained in roughly the same time with only 3 GPUs[LRH+21] That makes this architecture highly relevant to the research done in this thesis. In particular, the speed of computation makes it ideal for reinforcement learning, where the speed of computation is often the limiting factor for models since data scarcity is not an issue.

For a description of the model architecture and training process we refer to [LRH+21]. We discuss here only our specific use of the architecture for solving DDxGym, and the results we achieved.

#### Training

As a tokenizer we use the medically trained tokenizer of `BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext`. This gives the fruitfly a strong biomedical prior. With such a large vocabulary how-

ever the model consists of roughly 32 million parameters. It is therefore quite a bit larger than the 13 million parameter big `BERT-small` we used for other experiments. Because of its simple 1-layer architecture it is however still massively faster to compute. For even faster computation and a smaller model footprint this vocabulary could be optimized and trimmed,or the number of kenyon cells reduced, but we see this as a very minor optimization for our use case.

We calculate word frequencies and pre-train it on PubMed[SAB+18] as per [LRH+21]. During Online Reinforcement Learning we then keep updating the word frequencies in each training step from the observations. This ensures that the word frequencies match the data that is actually encountered during the interaction with the environment as the agent learns. Additionally, we minimize the Energy function described as the objective function in [LRH+21] as an auxiliary objective to the reinforcement learning loss. This serves a similar function to the masked language modelling objective we use for the transformer models.

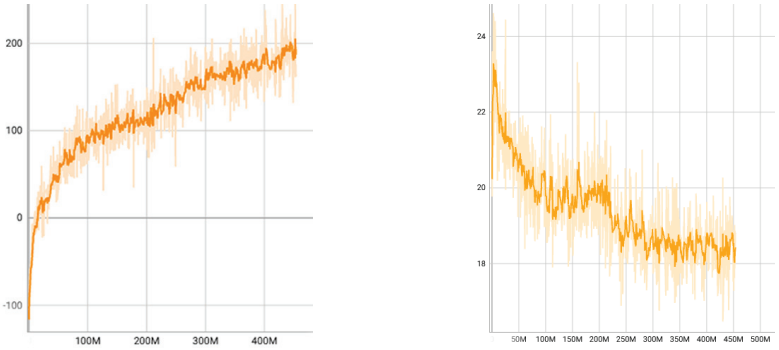



**Figure 5.8.** Result of training the fruitfly architecture in the DDxGym environment. Left: Mean Episode Reward per Step, Right: Mean Episode Length.

**Results and Analysis**

Even on substantially less powerful GPUs (P100 with 9.3 TFlops vs. A100 with 19.7 Tflops we used for the other experiments) the fruitfly trains multiple times faster than the small transformer model. While the environments and combination of different losses lead to a substantial overhead, the fruitfly architecture still shines in per-step speed. The results of our experiment are shown in Figure 5.8.

Unfortunately we see mediocre results of a mean reward of about 200, and a mean length of 18. Even after a long-running training of over 400M steps the model does not stagnate and converge, however. Higher learning rates led to too much instability, but these results show promise for even better results with longer training or a different set of hyperparameters. We do not pursue this further at this time however, since the main object of this thesis is the improvement of the transformer architecture.

In summary, the fruitfly architecture represents a promising alternative to transformer language models, even in the medical domain. While it didn't manage to reach the same performance as the transformer in our experiments, nor in the original paper, its main advantage is speed. For applications where that speed is of particular performance, either because quick reaction times are paramount, or because computational power is especially scarce, this architecture might be the more suitable one. We leave further exploration of this architecture for future work.

## 5.7 Discussion and Analysis

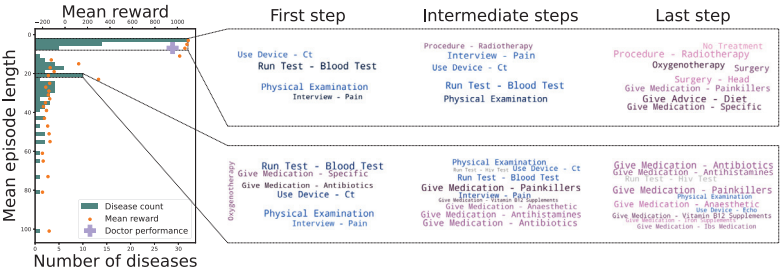



**Figure 5.9.** Inference on 5000 episodes with the best *T+MLM model*. *Left*: distribution of diseases across episode lengths. For 50 diseases the agent solves the environment in under 6 steps on average. Intuitively, a high mean reward corresponds to a short episode length (orange markers). Doctor performance on 16 diseases is shown by the violet cross.*Right*: we qualitatively examine the distribution of actions of the episodes solved under 6 steps (top), and in $[19, 20]$ steps (bottom). For the solved diseases the agent learns to uncover symptoms initially (blue actions) and then follows these with treatments(magenta actions).

We want to further analyze the trajectories traversed by our reinforcement learning agent, to better judge the model's suitability to these problems. To that end, we run 5000 episodes of inference with a checkpoint of T+MLM that achieves the highest mean reward. We analyze the reward and disease distributions with respect to episode lengths. As shown in the previous results, the episode length here serves as a proxy for the reward that can be discretized and is therefore suitable for distributions. Further, we analyze action distributions, with respect to successful and unsuccessful episodes, and the time development within an episode. Lastly, we analyze the relation between the difficulty of particular diseases and their examination overlap with each other.

In Figure 5.9 *Left* we present the distribution of the distinct diseases with respect to the episode lengths. There are 50 diseases for which the episodes last 6 steps or fewer. This is the largest group of diseases where the agent also achieves the largest reward, slightly over 1000 and very

close to the theoretical maximum for our environment. We denote these diseases as *successful*. We note that for diseases longer than that the reward values are mostly negative with a few outliers. We render these diseases as *unsuccesful*.

In Figure 5.9 *Right* we focus on the action distributions for the episodes of these disease groups throughout the agent's interactions. These are qualitatively shown as word clouds with the frequency of the actions mapped to the text size. We make the distinction between act that uncover symptoms (blue) and actions of treatment (magenta).

We examined more closely the group of *successful* diseases, and the largest group of diseases with negative mean rewards. Namely, the groups with episode length of less than 6 steps and those with a length $\in [19, 20]$.

## Successful diseases

We expand on the action distributions of this group of diseases at the top row of Figure 5.9 (Right). The agent learns to uniquely use examinations in the first step (only blue actions), while in the intermediate steps the agent continues on trying diagnostic actions to finally treat the main symptom of the disease in the last step of the episode (only magenta actions). This means for these 50 diseases the agent learns medical trajectories of examining symptoms, considering diagnostics, assigning diagnoses and proposing treatments in six steps or fewer. We show one such successful trajectory of our agent for liver cancer in Figure 5.1a.

## Unsuccessful diseases

In contrast, for the second group of diseases (negative mean reward, second row of Figure 5.9), the agent is not successful at uncovering nor treating the disease. We highlight how there's no discrimination by the agent of diagnosis and treatment actions in any of the steps (both magenta and blue actions are present).

## Action Discrimination in Successful Diseases

We find remarkable the fact that the agent learns to discriminate between diagnostic and treatment actions since they are not explicitly distinguishable within the action space. It is also noteworthy that the agent learns to treat the disease only once the main symptom is uncovered, this is evident since the intermediate steps involve to a great extent only diagnostic actions. We believe that this *awareness* regarding the action space and episode state is enabled by the coexisting MLM objective while training, since via this mechanism recall of past episodes is tightly coupled to the reward. We find that this setup is similar to model-based RL, but with the transformer at the center being both the policy and modelling the environment, just not its transition dynamics.

## Examination Overlap

Focusing only on examination actions that uncover the main symptom, we construct a disease-pairwise comparison of the episode lengths and examination-action overlaps for the simulated episodes. We group the lengths with respect to the size of the intersection of the examinations for the disease pairs, we denote this the *examination overlap*. Figure 5.10 *Left* shows the distributions of the mean episode lengths for these overlap groups. In parallel, we examine the behavior of the mean reward for the same groups in Figure 5.10 *Right*. We note that the groups of disease-pairs with higher overlap present both lower episode lengths as well as higher rewards. This is consistent with the diseases that the agent learns to diagnose. As the examination overlap decreases, the distributions of these pairwise comparisons shift to a higher episode length, consequently, a lower reward, and a noticeably higher variance for both. We believe that this relation of the episode length/reward and main-examination overlap is coherent with the idea that the agent learns to use diagnostic actions that are more general for all the diseases, i.e. the ones that on average would help in uncovering the main symptoms of as many diseases as possible, de-prioritizing actions that work only for a single disease. We have similar findings with doctors.

### Human Expert Trajectories

We sample the performance of a medical doctor in DDxGym for 16 diseases. We mark the mean reward and mean episode length captured in these episodes in Figure 5.9. This experiment shows a medical professional can achieve almost ideal rewards, even without prior experience with the environment. The doctor immediately identified and used many of the same actions that were also discovered by our best agent as being the most generally applicable, such as running a generic blood test, and different types of physical examinations. However, as episodes went longer, the doctor was much more capable than our agent, of choosing procedures that complement each other to further narrow the set of possible diseases. The doctor faced the greatest challenge, and even failed in one episode, with diseases outside of their specialty, or diseases that they haven't encountered before. We motivate further research in our methods to support medical practitioners, particularly with these diseases.

In summary, our reinforcement learning agent struggles in particular with diseases that have very common symptoms, but require unique examinations to be diagnosed. However, for the large group of diseases that the agent handily solves, the trajectories are very similar to the trajectories of an actual doctor, which the age giving further affirmative evidence to our research question.

## 5.8 DDxGym Demonstrator

We have developed a further demonstrator for our work on ´DDxGym: Online Transformer Policies in a Knowledge Graph Based Natural Language Environment´ [WFL+23]. It is available at `https://medicalrl.demo.datexis.com/` This demonstrator allows the user to interact with the DDxGym environment in the same fashion as the reinforcement learning agent does. Further, it allows for debugging by optionally exposing additional information about the episode, the underlying disease, and the sub-graph of the knowledge graph that is relevant to the current patient. This is the same demonstrator that was used in the qualitative evaluation of our environment with the medical professional. Like during reinforcement learning training, the demonstrator generates one patient with one disease

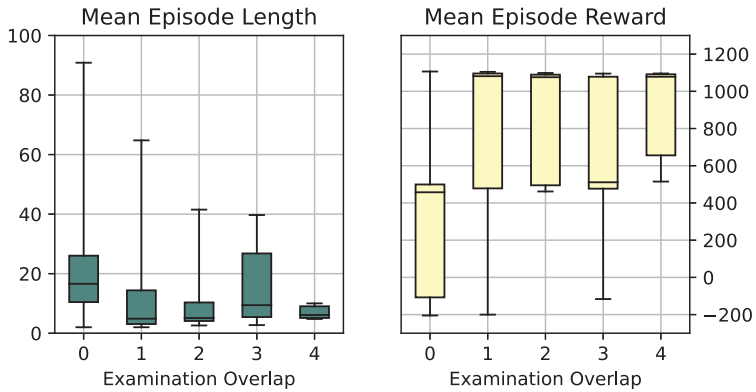



**Figure 5.10.** Distribution of the mean episode lengths (left) and rewards (right) with respect to the overlap in examination actions that uncover the main symptom. We note that for diseases with higher overlap, the episode lengths tend to be shorter and, consequently, rewards higher. In contrast, the diseases with low examination overlap remain challenging for the agent with longer episodes. Thus, the agent prioritizes diagnostic actions that are the most broadly applicable.

per disease and reveals one symptom in the first observation. From there, the user interacts with the environment by choosing actions via the input field. Figure 5.11 shows the UI of this demonstrator.

## 5.9 Limitations

While our goal was to model the process of differential diagnosis as realistically as possible, we had to make simplifications due to resource constraints. Most of these can be addressed and improved upon by the collection of more data, or data with a higher granularity, that can then be used in the DDxGym environment.

Patient representations are complex and multimodal, including beyond text laboratory results and imaging data. However, the current version of the environment focuses solely on the text modality. Generating this multi-modal data is not trivial, and further labelling would be needed.

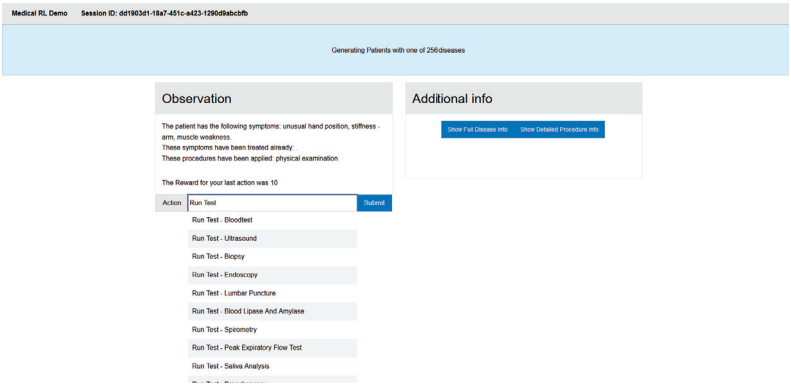



**Figure 5.11.** Overview of the DDxGym demonstrator interface. The observation mirrors what the RL agent would see, and it includes the reward of the last step. An action is chosen with the help of an autocomplete feature.

Furthermore, a transformer encoder alone would not be able to encode all of that data, and other models, e.g., a CNN for image data, would be necessary.

Disease-disease interactions are complex to model, therefore, the environment currently does not account for comorbidity.

Thirdly, assigning risks and costs to different examinations and treatments is a major feature of such a system that is yet outside the scope of our research. Access to powerful diagnostic tools like CT and MRI scans is often very limited or have a long waiting list, and invasive procedures like surgeries can involve high patient risk or discomfort. It is therefore desirable that a policy is learned, that not only finds quick trajectories to diagnosing and curing a patient, but does so while keeping in mind patient comfort and costs, only choosing certain actions when they are absolutely necessary. To that end, additional labels for the procedures would be required, and the reward structure updated accordingly.

Lastly, the observations generated by our RL environment lack the fidelity, variance and some of the technical language that is common in real world EHRs. They are challenging to generate automatically, and

our environment builds the foundation for that in a basic way. Through templating, and the use of powerful generative models this environment can be made even more challenging. This would also result in agents that can be more easily adapted to the real world.

## 5.10 Summary

In this chapter we addressed the research question "Is RL a suitable alternative to supervised learning in the Differential Diagnosis scenario?".

To that end, we presented DDxGym, a novel text-based reinforcement learning environment that models this challenging medical task. In addition, with the help of medical professionals, we created a novel medical knowledge graph with 111 diseases and their symptoms, related procedures, and their interactions. Further, we developed a novel RL approach using a masked language modelling objective that runs concurrently to modelling the policy, which addresses problems of learning instability and the overall performance of transformer language models in online RL. This approach significantly outperforms reasonable baselines. Lastly, our qualitative analysis with medical professionals shows that our system learns meaningful medical trajectories and general diagnostics early and uncovers main symptoms for many diseases.

# Part III

# Closing Discussion

Chapter 6

# Review of Conducted Research

## 6.1 Review of the Research Questions

In this chapter we will revisit and reexamine the research questions that were posed in the beginning of this thesis. In the light of the research contributions discussed in the previous chapters, we will evaluate and summarize the concrete findings to these questions.

### Research Question 1: Do Transformer Models contain the NLP Pipeline in their Layers?

In our journey to better understand Transformer models, we've taken a slightly different approach than what was previously done. Instead of primarily looking at how attention works in these models, we explored how different the encoder blocks in their entirety behave in these big networks. In order to gain a deeper understanding of this powerful and popular architecture we conducted both a qualitative and quantitative analysis targeting each layer of these networks. We focused our analysis on the downstream task of question answering because of its complexity and compositional nature. There, we made a few interesting findings and these findings are largely supported by both types of analysis. Each layer acts differently and exhibits stronger performance in certain tasks than others. We discover 4 distinct phases that layers can be grouped into: Topical Clustering, Connecting Entities with mentions and attributes, matching questions with supporting facts, and finally, answer extraction. The first two of these phases can be understood as task agnostic phases, which most closely align with steps in the traditional NLP pipeline, e.g. Named Entity recognition in phase 2. The third and fourth phases however are

specific to the question answering task, and are likely to differ for other downstream tasks. Similarly, for the quantitative analysis, we find that earlier layers perform better on simple probing tasks such as Named entity labelling, while late layers excel in the more complex tasks like supporting fact labelling.

While our findings give some clues about how Transformers might work, these models are still very complex. Additionally, some of the results of the quantitative and qualitative analyses are open to interpretation, so no fully conclusive answer can be given. However, the consistency in our findings across different types of analysis, as well as different models and model size give us confidence in our findings. These findings highlight the importance of targeting specific parts of a model for transfer learning. For example, it might be prudent to apply transfer learning only to the task specific layers relating to phase 3 and 4, in order to not destroy the more universal embedding built in phases 1 and 2. We take a more nuanced view to this however in our study of research question 2.

## Research Question 2: Can over parameterized models be improved through Knowledge Graph Completion Retraining?

Using the knowledge we gained throughout our study of research question 1, we then sought to improve on the generic transfer learning approach of pre-training and fine-tuning. We identified that for niche domains like the clinical domain limited data, domain-specific terminology, distinct relational knowledge, and catastrophic forgetting are key challenges that are difficult to overcome with regular pre-training. These hold particularly true in applications where efficiency is of high importance, which is the norm for this domain. To address these challenges we developed a novel transfer learning approach that includes an additional training step based on targeted knowledge graph retraining. Using knowledge graphs in this secondary training allows us to increase the amount of usable training data by utilizing structured data, and domain-specific knowledge graphs when chosen carefully contain exactly the specific terminology and knowledge that generic language models lack from their pre-training. Further, our learning approach utilizes ideas from model compression research in order to target the training on a very granular level to only attention heads that

are underutilized. This addresses the catastrophic forgetting problem as evidenced by results in the GLUE benchmark, while also creating more parameter efficient models as we improve models without increasing the amount of parameters.

Our research confirmed that even relatively small transformer models exhibit a great deal of over parameterization. When applied to the clinical domain, our specialized retraining led to significant improvements in model performance. This was especially pronounced in the more challenging Zero-Shot setting. Our research thereby highlights a promising avenue for the efficient adaptation of large models, striking a balance between domain specialization and broad linguistic comprehension.

However, utilizing new types of data is not the only avenue worth exploring when improving on transformer language models. In the study of research question 3 we explore an entirely different learning paradigm to further break with the traditional fine-tuning approach.

### Research Question 3: Is RL a suitable alternative to supervised learning in the Differential Diagnosis scenario?

In the field of machine learning, the dominance of supervised and semi-supervised methods is evident, given their remarkable efficacy across a diverse range of tasks. However, these approaches may yield models prone to adversarial attacks and making decisions for the wrong reasons. Further, there are applications that profit from models proficient in not only ad hoc decisions, but sequential decision-making and foresight. One such application is differential diagnosis. In order to solve DDx, a model needs to interact with a patient, repeatedly change its representation given new information, and gather new information by conducting sensible examinations on the patient. To that end we explored RL as an alternative learning paradigm. We developed the first comprehensive differential diagnosis RL environment, in which an agent diagnoses and treats a patient. We collected data for this environment from medical resources and employed medical professionals for its verification. Then, we solved the key problem of instability in policy learning, when using transformer models as a policy. We did this by continuous parallel MLM training.

Our best agent was able to solve most of the diseases in our dataset

with near perfect efficiency. Further, the medical trajectories formulated by the RL-trained transformer agent compared favorably with human medical professionals for a majority of diseases, in particular those with distinctive symptoms and widely applicable diagnostics. This kind of human-like decision-making is exactly what we predicted RL to be able to achieve, and highlights the main benefit over supervised learning. However, challenges arose in episodes where diseases presented with generic symptoms but require unique diagnostics for accurate identification. Here, the human doctor maintained a substantial edge over our RL actor. Still, given the breadth of diseases where the RL approach exhibited strong performance, our research shows that RL presents a promising alternative to traditional supervised learning. Future work will need to analyze more closely the trajectories traversed by our agent, especially in complex cases, to see both where it can be improved, but also where potentially medical professionals can benefit from the unique perspective of our agent.

## Overall Evaluation

The research presented in this thesis set out to accomplish two objectives: To further the understanding of transformer language models, and to use that knowledge to address the key challenges in their efficient transfer learning. Our first research question led us to conduct a qualitative and quantitative analysis, which illuminated the models' internal processes and transformations. Armed with that knowledge we explored two distinct avenues of improvement for these models in the context of low data scenarios: For our second research question we integrated structured domain-specific data into generic models, targeting only specific attention heads, developing an efficient training approach utilizing previously unused data. For our third research question we address the data limitation problem by generating data via a RL environment. We solve the key problem of learning instability in such environments using an auxiliary MLM objective, and find that our agent learns comparable trajectories to medical professionals.

## 6.2 Limitations of Presented Work

This thesis, while aiming to contribute valuable insights into transformer models and their domain adaptation, inevitably encounters several limitations that merit careful consideration. We summarize here once more the limitations concerning the research presented.

Firstly, the scope of the research is constricted to a narrow set of transformer models. While insights drawn from these models may be transferrable to other models with similar architectures, this transferability cannot be guaranteed.

Specifically, at the time of conducting this research, there were constraints related to accessibility and hardware that prevented the inclusion of some of the largest transformer models available in the field, such as Llama, MegatronLM and similar Large Language models. Even the smallest of the Llama models for example has around 70 times more parameters than the BERT-base-uncased model that has found great use in this thesis. And the complexity of using such a large model increases exponentially, not linearly. This is due to the greater number of parameters not only leading to longer training and inference times, and also require more data to be trained optimally, but simply loading all parameters on GPU hardware with limited memory is a challenge. Here tricks such as Model sharding and quantization become paramount. Even using such tricks however, it proved to be intractable to do research on that scale for us. As highlighted in the beginning of the thesis, the research on these large and powerful models is worryingly constricted to only a small set of companies with the required hardware budget. This means that this thesis lacks potentially valuable insights and understandings that might be unique to these larger models which have recently become incredibly popular. They have already exhibited a number of different abilities that were previously not thought possible. Still, they are transformers at heart, therefore much of this research should be transferable to them in one fashion or another.

Another limitation resides in the reliance on limited and imperfect datasets. The datasets employed in this thesis are inherently constrained, able to model only a subset of the myriad of real-world problems and scenarios encountered in practice. Consequently, the findings and conclusions

drawn from these datasets need to be interpreted in context, considering the gap between the controlled environments of the study and the multifaceted, dynamic nature of real-world applications. One big difference, especially in the medical domain, is the noisiness and sparseness of data. All the datasets used in this thesis are curated and cleaned in some fashion. With no real general standard for how to write Doctor's letters that build a large portion of patient representation, that is a nonexistent luxury in real world applications. Further, limiting our research to the narrow field of Natural Language processing, while somewhat necessary for a thesis such as this, is entirely inadequate to address complex problems in the medical field. Those problems require a much more holistic view on patients, seamlessly integrating different types and sources of data in a dynamic manner.

The results presented in this thesis are also not entirely indisputable, especially in the interpretation of results. While we can give some suggestions on what the difference phases in the layers of transformer networks could mean, how KIMERA is able to lead to such significant performance increases, and why our Reinforcement Learning Agent learns the trajectories it does, they are still subjective interpretations. These models are complex and abstract. They follow simple mathematical processes. But especially with powerful generative models, it is very easy to anthropomorphize them, and attribute to them feats of thinking, dreaming, and other human abilities. This is a very dangerous act, since it leads us to trust such models more than we should. We simply do not know yet how much these models actually understand language, whether their abstractions and generalization are meaningful, or whether they are just stochastic parrots[BGM+21]. Therefore, the interpretations of our results we give are to be taken with a grain of salt, and more evaluations and analyses have to be done to be able to confirm them.

Specifically for our KIMERA approach a great limitation is also that it is contingent upon the matching between sets of models, knowledge graphs, and downstream tasks. For effective application, the models and knowledge graphs need to complement each other in a manner that aligns with the requirements and objectives of the downstream task at hand. Determining this alignment and compatibility is a non-trivial endeavor, often nuanced and context-dependent, introducing another layer of complexity

and limitation to the practical application of the KIMERA approach.

Finally, the results derived in Chapter 5 are constrained by the absence of openly available benchmarks for the evaluation of real-world Differential diagnosis problems in online RL contexts. Without these benchmarks, assessing the real-world applicability and efficacy of the DDxGym approach is challenging, making it difficult to gauge the true value and impact of the findings in practical, applied settings. This lack of benchmarks introduces ambiguity in the evaluation process, necessitating future work to validate and corroborate the findings under different conditions and with more robust evaluation metrics.

Lastly stands a limitation in regard to our biases, and the actual application in the medical field. While there was some influence of medical professionals at every stage of this thesis, it was often indirect. It has to be stated that the research presented here is NLP research, not medical research. There might therefore be a mismatch between patient's needs, optimal healthcare outcomes, medical professional's needs, and the approaches presented in this thesis. It will take a further interdisciplinary effort to transform the results achieved in this thesis into something that can be set loose in the real world.

In sum, while this thesis advances the field's understanding of transformer models and their applications, and provides novel approaches for transfer learning, it is important for readers to approach the findings critically, and consider the outlined limitations as they interpret and apply the research results. More importantly though, these limitations also reveal areas where future research can contribute, offering pathways for further investigation and study in the field.

Chapter 7

# Outlook

## 7.1 Business Perspectives

Transformers and large language models (LLMs) have undeniably made a significant imprint on the landscape of technology already, being applied to a plethora of applications across various domains. From search engines, sentiment analysis, content moderation, and recommendation systems to sophisticated advertising algorithms, these models have enhanced the efficiency and accuracy in tasks that require understanding of human language. The current trajectory in the development of these models seems to favor scale — both in terms of model size and the volume of data they are trained on, aiming for unprecedented levels of performance and capability.

In particular Large Language Models with billions of parameters and popularized by ChatGPT have recently led to a new surge in applying NLP models to business tasks. Through their powerful generative capabilities they can be used for a wide variety of writing tasks such as writing code, database queries, website content, and more. However, some skepticism is required in the usage of these models still. They remain black box models with difficult to determine biases and actual knowledge, and they introduce even further difficulties in their evaluation and benchmarking, since finding accurate metrics for generative models is notoriously difficult.

This development marks an entirely new way that such models are utilized by businesses. With the hardware and data requirements being so immense for these powerful models, it becomes most efficient to simply rent access to the models that already exist from one of the very few providers of them like OpenAI, rather than training specialized in-house models. If data privacy permits, companies might even be inclined to send

their own data to these providers with the promise of more specialized models. While that is a straight-forward and easy solution, it comes with its own problems. A company would lose out completely on the control of the models they utilize, and their data might even become accessible to competitors depending on how the provider handles the data and training.

Therefore, despite the surge towards developing colossal models, there is a discernible need and growing demand in niche domains for models that are smaller and more efficient. This need arises from the challenges like data availability, computational resources, and hardware limitations, that have been previously discussed in this thesis pertaining to fields with data privacy requirements like healthcare. In these critical domains, the requirement is not always for a model that 'knows' more but for one that understands and operates efficiently within the specific context, providing reliable and accurate insights even when data is sparse or hardware is not at the cutting edge.

Even within these constraints, the advancements in large models are not irrelevant though. Techniques like model compression and adaptation allow for the extraction of significant value from large pre-trained models, effectively distilling their capabilities into smaller, domain-specific counterparts. Through processes like knowledge distillation, pruning, and quantization, large models can be compressed to fit the hardware and data limitations of niche domains without substantial loss of performance. Similarly, model adaptation allows for these smaller models to be fine-tuned and specialized for tasks prevalent in these domains, thereby optimizing their functionality and efficiency for domain-specific applications. Two approaches for this have been developed in this thesis.

Within the health sector, a domain where accuracy and reliability are paramount, these adapted models can play a crucial role. Whether it's assisting in diagnostics, streamlining patient care through intelligent triage systems, or aiding in the discovery and research of novel treatments and medications, the potential applications are vast and significant. For instance, in tasks like Differential Diagnosis (DDx), where the identification of diseases is made through symptom analysis, these models can offer invaluable support in quickly and accurately narrowing down potential health issues, or selecting relevant patient cohorts, thereby assisting healthcare professionals in making informed decisions.

Beyond improving efficiency and accuracy, these models can also provide 24/7 service, and handle large volumes of requests and data simultaneously. For businesses, this translates into not only better service provision but also the possibility of scaling operations in ways previously unattainable, unlocking new potentials for growth and service delivery in the digital age.

In summary, while the trend towards larger models continues, the business perspective for their application is in our opinion two-fold. For a wide variety of general tasks these immense and powerful models will be deployed by a very small number of companies. Other businesses will have to contend with surface-level access granted by these providers. Especially in critical domains like healthcare however, it lies in adeptly leveraging their capabilities through compression and adaptation techniques. These practices allow for the deployment of efficient, capable, and reliable models that meet the specific demands and constraints of the domain, driving improvements in service provision, operational efficiency, and ultimately, business performance.

## 7.2 Future Work

The ever-growing realm of Natural Language Processing continues to be a cornerstone of artificial intelligence research, evolving in ways that few could have anticipated just a few decades ago. One clear trajectory, as evidenced by recent trends, is the scaling up of models. These behemoths, while impressive in their capabilities, are increasingly only trainable, or even efficiently usable, by large corporations such as Google, OpenAI, and Nvidia. With their near limitless access to state-of-the-art hardware, and large internal data sources these companies dominate the space, pushing forward the development of such massive models. This trend is likely to continue for some time, since at least so far there seem to be little diminishing returns on the benefits of bigger models, as long as enough data is fed to them.

**Efficiency**. This corporate monopolization presents a challenge for academic and independent research of smaller groups. The lack of comparable hardware, combined with restricted access to the internals of some of these proprietary models, means that universities and smaller research entities often find themselves at a considerable disadvantage. In light of these constraints, these entities have to shift their focus towards developing smaller, more efficient models. There is a silver lining here though, as this shift can lead to a more focused exploration of areas potentially overlooked by larger corporations. These include tackling problems associated with sparse data domains, less-resourced languages, less common data modalities such as structured data, as well as the 'soft-skills' of models, such as improving explainability, ensuring model accountability, and detecting or even removing model biases. The research presented in this dissertation has explored and discussed 2 promising avenues for (data-) efficient domain transfer, using structured data and RL respectively.

**Explainability**. In an age where AI's decisions can profoundly impact lives, especially in critical sectors like healthcare, finance, and justice, the importance of explainability and accountability cannot be overstated. These are not just academic pursuits but ethical imperatives. The coming years should, and likely will, see a concerted push towards making models not just performative but also interpretable, with clear mechanisms to hold them accountable for their decisions. The research presented here has taken a first step in this direction by furthering the understanding of the internal processes and transformations that happen within deep transformer models.

**Transfer Learning**. Domain adaptation and transfer, despite recent advances, remains an area of research with yet more to be explored in our opinion. Adapter models, for instance, offer an interesting promise of extracting value from larger pre-trained models without the associated computational overhead. As industries and academia both continue to grapple with the challenges of domain-specific applications, such modular approaches might gain prominence, enabling more flexible and efficient solutions to domain-specific challenges. These modular approaches are a promising alternative to the approaches explored in this thesis, or might

even be used in conjunction with them. For example, one could envision a "knowledge graph module", that is trained in the fashion of KIMERA, that sits on top of a generic model and acts in concert with other such modules, trained on different data or for different purposes. This however goes beyond the scope of this dissertation and is seen as future work.

**Novel architectures and Learning Algorithms**. Looking further into the future, eventually there is likely to be a plateau in the scalability of models. There will be a physical and computational limit beyond which merely adding more parameters will yield diminishing returns that will eventually outweigh the costs. This will necessitate the exploration of entirely novel architectures and alternative learning paradigms. Reinforcement Learning (RL) offers one such avenue, though it brings its own set of challenges, like the need for vast amounts of generated samples and training steps, and the issue of volatile gradients. Our research has shown however, that it is a promising alternative to supervised learning for niche domains where data is limited. And already RL is finding application with large models for example in the form of Reinforcement Learning from Human Feedback(RLHF) strategies and in our opinion this trend will continue.

**Evaluation and Benchmarking**. Furthermore, the accurate evaluation of models will become paramount. With generative models' capabilities of creating entire datasets, the tasks of benchmarking, ensuring dataset quality, and identifying relevant metrics will become challenging, yet essential undertakings. Even for human created benchmarks there is research that proves their critical flaws and biases, that are exploited by conventional models. This is in part what lead us to our research questions 1 and 3. There is no reason to believe that these challenges don't arise in generated dataset. These challenges make it difficult to trust performance evaluations of strong models in particular, as they might simply be very proficient in cheating or breaking the benchmark in unforeseen ways.

In conclusion, there is a vast field of challenges in NLP that have to be solved in the future, and it is these very challenges that offer exciting opportunities for innovation, discovery, and research. As this thesis has

demonstrated, by revealing limitations and strategically choosing areas of maximum potential impact, the future of NLP can be as promising, if not more, than its already momentous past.

Chapter 8

# Conclusion

The research presented in this thesis collectively targets the improvement of transformer models for applications in niche and small data domains, with a specific focus on the medical domain. Each chapter contributes to an overarching goal: to develop methodologies that enable transformer networks to operate effectively in scenarios where traditional training approaches, characterized by vast amounts of data and compute resources, are infeasible. In particular, we have developed approaches that exploit data sources which are not commonly used, and are able to improve models with only little data available.

Initially, we provide a foundational analysis of the hidden layers within transformer models. This analysis into whether Transformer models encapsulate the NLP pipeline within their layers suggests that the layers of Transformers are modular in their make-up, and may perform tasks analogous to traditional NLP pipeline stages. While the complexity of transformer models makes this insight less than definitive, it highlights a promising new avenue for interpretability of these models. By conducting both qualitative and quantitative experiments, this part of our work reveals how different layers process and represent information. This exploration is crucial for understanding the internal mechanics of transformers, and in particular how different parts of the architecture manipulate language. Insights from this analysis importantly inform our development of approaches for model refinement and adaptation, ensuring that our modifications or enhancements are grounded in a solid understanding of the underlying transformer architecture. This sets the foundation of our further research.

Building upon this understanding, we introduce a novel transfer learning method, KIMERA, which incorporates knowledge graphs for domain-

transfer and employs a model compression method for targeting of underutilized attention heads. This approach leverages the structured information within knowledge graphs to enrich the model's context and understanding of medical terminologies and relationships, which is often absent in nonmedical training corpora. The targeting of attention heads, informed by the initial research's insights into layer functionalities, helps in avoiding catastrophic forgetting, and reduces the computational load of fine-tuning without sacrificing performance. We have found, that even small transformer models are indeed over parameterized even for challenging downstream tasks, and that this over parameterization can be effectively exploited. Knowledge Graphs have here been proven to be a valuable additional resource to ease the data limitation problem. This method addresses the data scarcity problem specifically by making use of rarely used structured data as an additional source of training data beyond the downstream task samples.

We then extend the concept of using knowledge graphs for domain-transfer by introducing a reinforcement learning approach based on an environment that is generated from a knowledge graph. In this environment, an agent learns to perform the complex medical task of differential diagnosis, akin to a human doctor. This approach not only tests the practical applicability of transformer models in real-world tasks but offers a further solution to the data scarcity problem by generating near infinite training trajectories from even a small knowledge graph. The suitability of Reinforcement Learning as an alternative to supervised learning in Differential Diagnosis scenarios was confirmed through these experiments. Transformer-based RL agents, stabilized by an auxiliary MLM objective, demonstrated capabilities akin to human doctors in diagnosing a variety of diseases. This suggests RL's potential as a viable alternative learning method in particular for medical, and possibly other dynamic and complex real-world scenarios.

Overall, we demonstrate a comprehensive strategy for enhancing the usability and domain adaptation of transformers in niche applications: starting from a fundamental understanding of the model's internal operations, through innovative adaptations using domain-specific resources like knowledge graphs, to practical applications in complex, real-world tasks.

In the current landscape of Natural Language Processing (NLP) and

its applications in the medical domain, the research presented in this thesis holds significant relevance both academically and practically. The foundation we have laid in understanding transformer models has already formed the basis for a number of different research efforts[RKR20; MRP+20]. Thus, we have contributed to the broader NLP discussion on model interpretability and efficiency. Understanding how these complex models process and represent language is essential, especially as their applications extend into high-stakes areas like healthcare. Further, as NLP continues to become a prevalent tool to solve real-world problems, the ability to efficiently adapt and apply these technologies in specialized, data-constrained environments like healthcare remains crucial. Our work addresses this need through a successive exploration of transformer networks, specifically focusing on enhancing their adaptability and utility in medical contexts.

Our research advances the integration of domain-specific knowledge through innovative transfer learning techniques. By incorporating knowledge graphs and targeted model adjustments, we address the challenge of data scarcity that often hinders the application of machine learning in fields like medicine. Our approaches not only improve model performance without the use of extensive data, but also, in part, reduce computational demands, making it feasible to deploy advanced NLP models in resource-limited environments.

Even as the field of Natural Language Processing (NLP) evolves, with developments pointing towards increasingly larger and more complex models such as GPT-5, the relevance of our research, which focuses on optimizing smaller transformer models for niche and small data domains like medicine, remains high. This is rooted in several key aspects of NLP research's trajectory and the unique challenges and opportunities presented by niche applications.

**Scalability and Computational Efficiency.** While the trend in NLP has been towards creating larger models, the scalability and computational demands of Large Language Models often limit their practical application, particularly in environments constrained by resources or those requiring real-time processing. Our research on small transformer models such as BERT-base addresses these challenges by increasing efficiency with optimized transfer learning approaches. Additionally, as models continue

to grow, the insights gained from making smaller models more efficient and capable could prove invaluable in scaling down larger models or in refining their architecture to be more computationally manageable without significant losses in performance. And as hardware improves, larger models become usable in more real-world applications, and can profit from the approaches developed in this thesis.

**Ethical and Practical Deployment.** When deploying models in fields like medicine, ethical concerns become increasingly important. For example, models that are not openly available, or which can't be run on local hardware for other reasons, might simply not be an option. Patient privacy concerns forbid the uploading of patient data, often even if anonymized, to a server of a private company which provides only few assurances in regard to how the data is handled. For such applications, small, local models will continue to be without alternatives. Further, Interpretability and explainability are of crucial importance in domains like medicine where models can have a direct impact on people's well-being. Our research touches on this aspect by providing a novel avenue for understanding transformer models. With large language model's architecture being largely similar to small transformer models this understanding can possibly transfer to these models as well.

**Domain-Specific Adaptation and Data Scarcity** Larger models, while generally more capable across a broad spectrum of tasks and domains, may not inherently possess the nuanced understanding required for every specialized domain. Our work, particularly in integrating domain-specific knowledge through knowledge graphs and adapting models via reinforcement learning to perform complex tasks like Differential Diagnosis, provides a blueprint for how both small and large models can be fine-tuned or adapted to meet specific real-world needs. These approaches ensure that as models grow, they can go beyond being jacks-of-all-trades and can excel in specific, critical applications as well.

Specifically, the problem of data scarcity remains pertinent. Many real-world applications suffer from a lack of large, annotated datasets necessary to train models like GPT-5 effectively. The methods we have developed utilize transfer learning and alternative data sources to significantly improve model performance in data-sparse environments. These strategies are directly applicable to large models as well, and can help them leverage

existing data more effectively and learn from smaller, domain-specific datasets.

**Evaluation** Accurately evaluating generative models is not a straightforward task. For generative tasks there are metrics such as BLEU and ROUGE, but even those often compare badly to human evaluation. When trying to evaluate generative models for classification and regression tasks, things become even more difficult [Rei23]. Generative Models' output is non-deterministic, and labels have to be extracted from it. One common strategy to address that is to sample outputs from the model multiple times, but of course this skews results, since in a real world application that might not be a sensible option. Regular classification models are also not afforded such lee-way in getting multiple attempts at getting the right answer. While evaluation of Large Language Models is a rapidly growing field, small transformer models will be more easily and accurately evaluated at least for the near future.

**Economical and Environmental Concerns** Even should ample training data and computational capacity be available, these resources are not free. The energy consumption of training Large Language Models can be immense, in particular during training. The actual power consumption of the pre-training of models such as GPT-4 is a closely guarded secret, but estimates from leaked information are in the single or low double-digit Gigawatt range[1] for one training cycle. That represents the average yearly consumption of hundreds or thousands of people in the developed world. With that also comes a hefty carbon footprint and environmental impact. While even small transformer models are not without issue in that regard, fine-tuning a small model is magnitudes more economical and environmentally friendly than the training of a Large Language Model.

While it is likely GPT-5 and other successors to the state of the art will continue to impress with their performance, that doesn't necessarily increase their applicability to every special domain, especially those which require more than simply good benchmark results. For the reasons previously outlined, small transformer models will stay relevant for these domains for the foreseeable future. These smaller models unfortunately do not exhibit the same amount of generalization prowess as large language

[1] https://tinyml.substack.com/p/the-carbon-impact-of-large-language

models. Therefore, approaches like the ones detailed in this thesis are crucial, to make the most efficient use of the data and hardware available to build domain-specific models. The integration of knowledge graphs in particular is valuable here since a lot of medical knowledge and knowledge in other fields is structured.

Of course, with the focus of this thesis being the use of knowledge graphs for domain adaptation, and other limitations that have been previously outlined, this thesis presents only one possible answer to the question of how to build efficient domain-specific models for real-world applications. As noted in chapter 2, there are several other approaches, which are different in nearly every way, but which aim to address the very same issues of data scarcity and lack of computational resources. It is yet unclear which of these approaches is the best, and the answer will largely depend on the concrete use case. Additionally, these approaches may be combined in order to reap the benefits of multiple such avenues. Research has to continue in all of these directions.

The ongoing exploration and refinement of approaches like the ones presented in this thesis will undoubtedly contribute to the reliability and robustness of machine learning applications in health-related fields, and with that the growing role of NLP research in medicine. However, data and computational efficiency are only a small part of what is necessary to allow deep learning models to be applied in critical domains. More research has to be done into decision legitimization, model bias, effective evaluation and similar topics, which are of equal importance to data- and computation efficiency for many real-world applications.

Overall, this thesis reflects one sub-movement in the NLP community towards creating more adaptable, transparent, and efficient NLP systems. By focusing on enhancing transformers for specific, critical applications, our research contributes to the ongoing effort to bridge theoretical machine learning advancements with impactful, practical applications, particularly in enhancing healthcare outcomes through technology.

In summary, the evolution of NLP towards larger models does not diminish the importance of research into smaller, more efficient, and domain-adapted models, which we have presented in this thesis. Instead, it highlights the necessity of such work to ensure the advancements in model size and complexity translate effectively and ethically into practical applications. Data scarcity and computational limitations in niche domains are challenges that are here to stay, and there is a continued need for models and paradigms that address them. This thesis offers possible solutions to these challenges. As NLP continues to develop, the principles and techniques explored in this thesis can provide guidance for making these powerful models accessible, and effective across a wide range of applications, particularly in high-stakes domains like healthcare.

Appendix A

# List of Utilized Software

## A.1 Programming

Python, PyTorch, pandas, scikit-learn, scipy, numpy, scrapy, ray, pytorch lightning, tensorboard, seaborn, matplotlib, labelstudio, git, github,

## A.2 Writing

LateX, Visual Studio Code, Grammarly, Overleaf

# List of Figures

3.1 Schematic overview of the BERT architecture and 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 span prediction head consisting of a Linear Layer and a Softmax Layer. We use the hidden states of each layer as input to a set of probing tasks to examine the encoded information. . 40

3.2 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 lost. . . . . 47

3.3 Probing Task results of BERT-large models in macro averaged F1 (Y-axis) over all layers (X-axis). Performance of 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. . . . . 48

3.4 BERT's Transformation Phases for the SQuAD example from Table 3.1. Answer token: Red diamond-shaped. Question Tokens: Orange star-shaped. Supporting Fact tokens: Dark Cyan. Prominent clusters are circled. The model passes through different phases in order to find the answer token, which is extracted in the last layer (#11). . . . . 50

3.5 BERT's Transformation Phases for the bAbI example from Table 3.1. The phases are equal to what we observe in SQuAD and HotpotQA samples: The formed clusters in the first layers show general language abilities, while the last layers are more task-specific. . . . . 52

3.6 Phases of BERT's language abilities. Higher saturation denotes higher accuracy on probing tasks. Values are normalized over tasks on the Y-axis. X-axis depicts layers of BERT. NEL: Named Entity Labeling, COREF: Coreference Resolution, REL: Relation Classification, QUES: Question Type Classification, SUP: Supporting Fact Extraction. All three tasks exhibit similar patterns, except from QUES, which is solved earlier by the HotpotQA model based on BERT-large. NEL is solved first, while performance on COREF and REL peaks in later layers. Distinction of important facts (SUP) happens within the last layers. . . . . 55

3.7 bAbI Example of the Answer Extraction phase in GPT-2. Both the question and Supporting Fact are extracted, but the correct answer is not fully separated as in BERT's last layers. Also a potential candidate Supporting Fact in ´Sheep are afraid of Wolves´ is separated as well. . . . . 57

3.8 BERT SQuAD example of a falsely selected answer based on the matching of the wrong Supporting Fact. The predicted answer 'lectures' is matched onto the question as a part of this incorrect fact (magenta), while the actual Supporting Fact (cyan) is not particularly separated. . . . . 58

3.9 BERT SQuAD example Layer 7. Tokens are color-coded by sentence. This visualization shows that tokens are clustered by their original sentence membership suggesting far reaching importance of the positional embedding. . . . . 59

3.10 VisBERT interface. Top: Basic information and data entry. Question, Ground Truth Answer, and question answers are shown and can be edited. Predicted answer by the model is shown as well. Bottom: Hidden state analysis with PCA. Slider controls which layer is shown. . . . . 60

4.1 **A)** KIMERA consists of three phases: **I** A transformer model is fine-tuned and a head-mask is computed by identifying redundancies. **II** The computed mask is then used in conjunction with a multi-task training based on knowledge graph completion. Finally, the model is fine-tuned on the target task. **III** The retrained model is fine-tuned on the domain-specific task to culminate the domain transfer. **B)** Examples of KG retraining tasks. **I** and **II** *Entity Prediction* with a Masked Language Modelling objective. **III** *Relation Prediction* with a multi-class classification objective, and **IV** *Triplet Classification* with a binary classification objective. . . 69

4.2 Analysis of $I_h$ and over parametrization before and after using KIMERA in clinical answer passage retrieval. **A)** Attention Mask generated in KIMERA Step I. **B)** Attention map $I_h$ of BERT-base before applying KIMERA. **B)** Attention map $I_h$ after applying KIMERA to BERT-base. . . . . 87

4.3 Statistics of shift in $I_h$ after applying KIMERA **A)** Shows a consistent increase of mean importance per layer. **B)** and **C)** show the effect of KIMERA split by retrained and frozen heads respectively. While retrained heads become significantly more important, the effect on frozen heads much less clear. . . . . 88

5.2 A history of observations for a full example episode of our best agent interacting with DDxGym treating liver cancer. . 98

5.3 Disease relations in the DDxGym knowledge graph. The same procedures might connect to different symptoms and therefore multiple diseases. . . . . 102

5.4 Model Architecture. For each environment step there are two forward passes over the same model. First, the observation $o_t$ is used to predict the value of the current state $V_t$, choosing the agent's next action $a_t$, as well as predicting the patient's disease for the T+PD baselines. In the second pass, the observation is masked and then used to train the masked language modelling objective. . . . . 102

5.5 Evaluation of various pre-trained transformer language models as policies in the DDxGym environment. Performance of the three models is comparable, while BERT-small operates at a substantially higher speed. To improve readability, exponential moving averages are presented with $\alpha = 0.85$. . . . . 109

5.6 Results in the DDXGym Environment with Project Hospital data, limiting the environment to different subsets of diseases. While the smallest set is very easily solved, even just 10 diseases lead to a considerable challenge for the transformer policy. . . . . 112

5.7 Comparison of five different baselines on the DDxGym environment. The transformer model with auxiliary masked language modelling objective (*T+MLM*) clearly outperforms other baselines, both in mean reward (left) and in learning stability. This is also noticeable in episode length (right), showing it manages on average to treat patients in the shortest amount of steps. . . . . 113

5.8 Result of training the fruitfly architecture in the DDxGym environment. Left: Mean Episode Reward per Step, Right: Mean Episode Length. . . . . 117

5.9 Inference on 5000 episodes with the best *T+MLM model*. *Left*: distribution of diseases across episode lengths. For 50 diseases the agent solves the environment in under 6 steps on average. Intuitively, a high mean reward corresponds to a short episode length (orange markers). Doctor performance on 16 diseases is shown by the violet cross.*Right*: we qualitatively examine the distribution of actions of the episodes solved under 6 steps (top), and in $[19, 20]$ steps (bottom). For the solved diseases the agent learns to uncover symptoms initially (blue actions) and then follows these with treatments(magenta actions). . . . . 119

5.10 Distribution of the mean episode lengths (left) and rewards (right) with respect to the overlap in examination actions that uncover the main symptom. We note that for diseases with higher overlap, the episode lengths tend to be shorter and, consequently, rewards higher. In contrast, the diseases with low examination overlap remain challenging for the agent with longer episodes. Thus, the agent prioritizes diagnostic actions that are the most broadly applicable. . . . 123
5.11 Overview of the DDxGym demonstrator interface. The observation mirrors what the RL agent would see, and it includes the reward of the last step. An action is chosen with the help of an autocomplete feature. . . . . 124

# List of Tables


3.1 Samples from SQuAD dataset (left) and from Basic Deduction task (#15) of the bAbI dataset (right). Supporting Facts are printed in bold. The SQuAD sample can be solved by word matching and entity resolution, while the bAbI sample requires a logical reasoning step and cannot be solved by simple word matching. Figures in the further analysis will use these examples where applicable. . . . . 42

3.2 Results from fine-tuning BERT on QA tasks. Baselines are: BIDAF [SKF+] for SQuAD, the LSTM Baseline for bAbI from [WBC+16] and the HotpotQA baseline from [YQZ+18] for the two Hotpot tasks. . . . . 44

4.1 Results across the four CAPR datasets using the Cross Encoder architecture(left) and four COP tasks(right). Top part shows scores for models based on BERT-base, bottom part scores for models on BioBERT. KIMERA improves on both BERT-base and BioBERT performance, with the exception of the LOS task. . . . . 78

4.2 Results of the GLUE benchmark, choosing the best of 10 seeds. KIMERA consistently outperforms BioBERT, and shows improvements over BERT-base in 3 tasks, having the highest mean score of tested models. . . . . 82

4.3 Overall Accuracy(%) across the different HellaSwag settings. BERT-base-uncased baseline result from HellaSwag Leaderboard[1]. No significant performance benefit of KIMERA is observed. . . . . 85

4.4 Quantitative evaluation of $I_h$. It leads to a significant increase in $I_h$ for previously unimportant heads and leads to a slight decrease of previously important attention heads. . 87

5.1 DDxGym knowledge graph statistics. The resulting environment actions are defined by the number of examinations and treatments, amounting to a total of 330. . . . . 100
5.2 *Acute pancreatitis* in our knowledge graph. There are four different symptoms. The disease identifier is the main symptom. Each of the symptoms has at least one *examination* that uncovers it. Not all of the non-main symptoms might actually affect the patient, this is governed by the *probability* field. They also might only appear after a few environment steps which is determined by the *onset*. While the goal is to treat the actual pancreatitis, the agent might find it useful to treat the *fever* and *nausea* if they are present, since their *severity* leads the patient to deteriorate more quickly. The symptom *jaundice* however, can not be treated directly as there is no *treatment* for it in our data. . . . . 103
5.3 Parameters for training transformer baselines. . . . . 106

# Bibliography


[A Z97] Lotfi A Zadeh. "Zadeh, l.a.: toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. fuzzy sets and systems". In: *ELSEVIER Fuzzy Sets and Systems* 90 (1997).

[AAG+20] Sebastian Arnold, Betty van Aken, Paul Grundmann, Felix A. Gers, and Alexander Löser. "Learning contextualized document representations for healthcare answer retrieval". In: WWW '20 (2020), pp. 1332–1343. DOI: 10.1145/3366423.3380208. URL: https://doi.org/10.1145/3366423.3380208.

[ABB+22] Michael Ahn et al. "Do as i can, not as i say: grounding language in robotic affordances". In: *Conference on Robot Learning*. 2022. URL: https://api.semanticscholar.org/CorpusID:247939706.

[AD19] Asma Ben Abacha and Dina Demner-Fushman. "A question-entailment approach to question answering". In: *BMC Bioinform.* 20.1 (2019), 511:1–511:23. DOI: 10.1186/s12859-019-3119-4. URL: https://doi.org/10.1186/s12859-019-3119-4.

[ALT+21] Neel Alex et al. "RAFT: A real-world few-shot text classification benchmark". In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*. Ed. by Joaquin Vanschoren and Sai-Kit Yeung. 2021. URL: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/ca46c1b9512a7a8315fa3c5a946e8265-Abstract-round2.html.

[AMB+19] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. "Publicly available clinical BERT embeddings". In: *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. Minneapolis, Minnesota, USA: Association for Computa-

tional Linguistics, June 2019, pp. 72–78. DOI: 10.18653/v1/W19-1909. URL: https://www.aclweb.org/anthology/W19-1909.

[APM+21] Betty van Aken, Jens-Michalis Papaioannou, Manuel Mayrdorfer, Klemens Budde, Felix Gers, and Alexander Loeser. "Self-supervised knowledge integration for clinical outcome prediction from admission notes". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2021, April 19-23, 2021, Volume 1: Long Papers*. Apr. 2021.

[AWL+19] Betty van Aken, Benjamin Winter, Alexander Löser, and Felix 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 '19. Beijing, China: Association for Computing Machinery, 2019, pp. 1823–1832. ISBN: 9781450369763. DOI: 10.1145/3357384.3358028. URL: https://doi.org/10.1145/3357384.3358028.

[AWL+20] Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. "Visbert: hidden-state visualizations for transformers". In: *Companion Proceedings of the Web Conference 2020*. WWW '20. Taipei, Taiwan: Association for Computing Machinery, 2020, pp. 207–211. ISBN: 9781450370240. DOI: 10.1145/3366424.3383542. URL: https://doi.org/10.1145/3366424.3383542.

[BAC+22] Christos Baziotis, Mikel Artetxe, James Cross, and Shruti Bhosale. *Multilingual machine translation with hyper-adapters*. May 2022. DOI: 10.48550/arXiv.2205.10835.

[BBB+93] Jane Bromley, James W. Bentz, Léon Bottou, Isabelle M Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. "Signature verification using a "siamese" time delay neural network". In: *Int. J. Pattern Recognit. Artif. Intell.* 7 (1993), pp. 669–688. URL: https://api.semanticscholar.org/CorpusID:16394033.

[BC19] Antoine Bosselut and Yejin Choi. "Dynamic knowledge graph construction for zero-shot commonsense question answering". In: *CoRR* abs/1911.03876v2 (2019). arXiv: 1911.03876. URL: http://arxiv.org/abs/1911.03876v2.

[BCP+16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. "Openai gym". In: *arXiv preprint arXiv:1606.01540* (2016).

[BDD+17] Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James R. Glass. "What do neural machine translation models learn about morphology?" In: *Proceedings of ACL 2017*. 2017.

[BF19] Ankur Bapna and Orhan Firat. "Simple, scalable adaptation for neural machine translation". 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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 1538–1548. DOI: 10.18653/v1/D19-1165. URL: https://www.aclweb.org/anthology/D19-1165.

[BGM+21] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. "On the dangers of stochastic parrots: can language models be too big?" In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 610–623. ISBN: 9781450383097. DOI: 10.1145/3442188.3445922. URL: https://doi.org/10.1145/3442188.3445922.

[BMR+20] Tom B. Brown et al. "Language models are few-shot learners". In: *CoRR* abs/2005.14165 (2020). arXiv: 2005.14165. URL: https://arxiv.org/abs/2005.14165.

[Bod04] Olivier Bodenreider. "The unified medical language system (UMLS): integrating biomedical terminology". In: *Nucleic Acids Res.* 32.Database-Issue (2004), pp. 267–270. DOI: 10.1093/nar/gkh061. URL: https://doi.org/10.1093/nar/gkh061.

[BPC20] Iz Beltagy, Matthew E. Peters, and Arman Cohan. "Longformer: the long-document transformer". In: *ArXiv* abs/2004_-.05150 (2020).

[BPT+04] Aziz A Boxwala, Mor Peleg, Samson Tu, Omolola Ogunyemi, Qing T Zeng, Dongwen Wang, Vimla L Patel, Robert A Greenes, and Edward H Shortliffe. "Glif3: a representation format for sharable computer-interpretable clinical practice guidelines". In: *Journal of biomedical informatics* 37.3 (2004), pp. 147–161.

[BRS+19] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. "COMET: commonsense transformers for automatic knowledge graph construction". 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*. Ed. by Anna Korhonen, David R. Traum, and Lluís Màrquez. Association for Computational Linguistics, 2019, pp. 4762–4779. DOI: 10.18653/v1/p19-1470. URL: https://doi.org/10.18653/v1/p19-1470.

[BWC+11] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. "Learning structured embeddings of knowledge bases". In: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*. Ed. by Wolfram Burgard and Dan Roth. AAAI Press, 2011. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3659.

[BYC13] J. Bergstra, D. Yamins, and D. D. Cox. "Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. Atlanta, GA, USA: JMLR.org, 2013, I–115–I–123.

[CBB+20] Souradip Chakraborty, Ekaba Bisong, Shweta Bhatt, Thomas Wagner, Riley Elliott, and Francesco Mosconi. "BioMedBERT: a pre-trained biomedical language model for QA and IR". In:

*Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 669–679. DOI: 10.18653/v1/2020.coling-main.59. URL: https://www.aclweb.org/anthology/2020.coling-main.59.

[CHH+23] Shiming Chen, Ziming Hong, Wenjin Hou, Guo-Sen Xie, Yibing Song, Jian Zhao, Xinge You, Shuicheng Yan, and Ling Shao. "Transzero++: cross attribute-guided transformer for zero-shot learning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.11 (2023), pp. 12844–12861. DOI: 10.1109/TPAMI.2022.3229526.

[CK18] Alexis Conneau and Douwe Kiela. "Senteval: an evaluation toolkit for universal sentence representations". In: *Proceedings of LREC 2018*. 2018.

[CLR+21] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. "Decision transformer: reinforcement learning via sequence modeling". In: *Advances in neural information processing systems* 34 (2021), pp. 15084–15097.

[Com94] Pierre Comon. "Independent component analysis, A new concept?" In: *Signal Processing* 36 (1994).

[CPL+22] Micah Carroll, Orr Paradise, Jessy Lin, Raluca Georgescu, Mingfei Sun, David Bignell, Stephanie Milani, Katja Hofmann, Matthew Hausknecht, Anca Dragan, et al. "Unimask: unified inference in sequential decision problems". In: *arXiv preprint arXiv:2211.10869* (2022).

[CTK+19] Yash Chandak, Georgios Theocharous, James E. Kostas, Scott M. Jordan, and Philip S. Thomas. "Learning action representations for reinforcement learning". In: *International Conference on Machine Learning*. 2019. URL: https://api.semanticscholar.org/CorpusID:59553460.

[DBH18] F. K. Došilović, M. Brčić, and N. Hlupić. "Explainable artificial intelligence: a survey". In: *MIPRO 2018*. 2018.

[DCL+19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/n19-1423. URL: https://doi.org/10.18653/v1/n19-1423.

[DES+15] Gabriel Dulac-Arnold, Richard Evans, Peter Sunehag, and Ben Coppin. "Reinforcement learning in large discrete action spaces". In: *CoRR* abs/1512.07679 (2015). arXiv: 1512.07679. URL: http://arxiv.org/abs/1512.07679.

[DGV+18] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. "Universal transformers". In: *Proceedings of SMACD 2018*. 2018.

[DHB+01] Paul A De Clercq, Arie Hasman, Johannes A Blom, and Hendrikus HM Korsten. "Design and implementation of a framework to support the development of clinical guidelines". In: *International journal of medical informatics* 64.2-3 (2001), pp. 285–318.

[DL15] Andrew M. Dai and Quoc V. Le. "Semi-supervised sequence learning". In: *Proceedings of NIPS 2015*. 2015.

[DSW+20] Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. "Adversarial and domain-aware BERT for cross-domain sentiment analysis". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 4019–4028. DOI: 10.18653/v1/2020.acl-main.370. URL: https://www.aclweb.org/anthology/2020.acl-main.370.

[DYY+19] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. "Transformer-xl:

attentive language models beyond a fixed-length context". In: *CoRR* (2019).

[ESM+18] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. "Impala: scalable distributed deep-rl with importance weighted actor-learner architectures". In: *International conference on machine learning*. PMLR. 2018, pp. 1407–1416.

[FAL17] Chelsea Finn, P. Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks". In: *International Conference on Machine Learning*. 2017. URL: https://api.semanticscholar.org/CorpusID:6719686.

[FDJ+15] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. "Retrofitting Word Vectors to Semantic Lexicons". en. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, 2015, pp. 1606–1615. DOI: 10.3115/v1/N15-1184. URL: http://aclweb.org/anthology/N15-1184 (visited on 11/28/2019).

[FJR98] John Fox, Nicky Johns, and Ali Rahmanzadeh. "Disseminating medical knowledge: the proforma approach". In: *Artificial intelligence in medicine* 14.1-2 (1998), pp. 157–182.

[FRS01] Karl Pearson F.R.S. "Liii. on lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2 (1901).

[GAL21] Paul Grundmann, Sebastian Arnold, and Alexander Löser. "Self-supervised answer retrieval on clinical notes". In: *CoRR* abs/2108.00775 (2021). arXiv: 2108.00775. URL: https://arxiv.org/abs/2108.00775.

[GMS+20] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah 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*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, July 2020, pp. 8342–8360. DOI: 10.18653/v1/2020.acl-main.740. URL: https://aclanthology.org/2020.acl-main.740.

[GMT+18] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. "A survey of methods for explaining black box models". In: *ACM Comput. Surv.* (2018).

[Gol19] Yoav Goldberg. "Assessing bert's syntactic abilities". In: *CoRR* (2019).

[HBE+24] Anson Ho, Tamay Besiroglu, Ege Erdil, David Owen, Robi Rahman, Zifan Carl Guo, David Atkinson, Neil Thompson, and Jaime Sevilla. "Algorithmic progress in language models". In: *ArXiv* abs/2403.05812 (2024). URL: https://api.semanticscholar.org/CorpusID:268358466.

[HGJ+19] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. "Parameter-efficient transfer learning for NLP". In: *CoRR* abs/1902.00751 (2019). arXiv: 1902.00751. URL: http://arxiv.org/abs/1902.00751.

[HR18] Jeremy Howard and Sebastian Ruder. "Fine-tuned language models for text classification". In: *CoRR* (2018).

[HRK+20] Martin Christian Hirsch, Simon Ronicke, Martin Krusche, and Annette Doris Wagner. "Rare diseases 2030: how augmented ai will support diagnosis and treatment of rare diseases in the future". In: *Annals of the Rheumatic Diseases* 79.6 (2020), pp. 740–743. ISSN: 0003-4967. DOI: 10.1136/annrheumdis-2020-217125. eprint: https://ard.bmj.com/content/79/6/740.full.pdf. URL: https://ard.bmj.com/content/79/6/740.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[HSL+20] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. "Poly-encoders: architectures and pre-training strategies for fast and accurate multi-sentence scoring". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=SkxgnnNFvH.

[HSW+21] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. "Lora: low-rank adaptation of large language models". In: *CoRR* abs/2106.09685 (2021). arXiv: 2106.09685. URL: https://arxiv.org/abs/2106.09685.

[Hug18] Huggingface. *Pytorch-pretrained-bert*. 2018. URL: https://github.com/huggingface/pytorch-pretrained-BERT.

[HVZ17] Dieuwke Hupkes, Sara Veldhoen, and Willem H. Zuidema. "Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure". In: *Proceedings of IJCAI 2018*. 2017.

[HZP20] Boran Hao, Henghui Zhu, and Ioannis Paschalidis. "Enhancing clinical BERT embedding using a biomedical knowledge base". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 657–661. DOI: 10.18653/v1/2020.coling-main.57. URL: https://www.aclweb.org/anthology/2020.coling-main.57.

[HZX+20] Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. "BERT-MK: integrating graph contextualized knowledge into pre-trained language models". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2281–2290. DOI: 10.18653/v1/2020.findings-emnlp.207. URL: https://www.aclweb.org/anthology/2020.findings-emnlp.207.

[HZY+18] Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. "FewRel: a large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ed. by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 4803–4809. DOI: 10.18653/v1/D18-1514. URL: https://aclanthology.org/D18-1514.

[JM09] Dan Jurafsky and James H. Martin. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, chapter 23*. Vol. 2. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, 2009.

[JPS+16] Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. "MIMIC-III, a freely accessible critical care database". In: *Scientific Data* 3.1 (May 2016), p. 160035. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.35. URL: https://doi.org/10.1038/sdata.2016.35.

[JSM+23] Albert Q. Jiang et al. *Mistral 7b*. 2023. arXiv: 2310.06825 [cs.CL].

[JW19] Sarthak Jain and Byron C. Wallace. "Attention is not explanation". In: *Proceedings of NAACL 2019*. 2019.

[KHK+20] Bosung Kim, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. "Multi-task learning for knowledge graph completion with pre-trained language models". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 1737–1743. DOI: 10.18653/v1/2020.coling-main.153. URL: https://www.aclweb.org/anthology/2020.coling-main.153.

[Kni] Will Knight. *Openai's ceo says the age of giant ai models is already over*. https://www.wired.com/story/openai-ceo-sam-altman-the-age-of-giant-ai-models-is-already-over/.

[KS20] Katikapalli Subramanyam Kalyan and S. Sangeetha. "Secnlp: a survey of embeddings in clinical natural language processing". In: *Journal of Biomedical Informatics* 101 (2020), p. 103323. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2019.103323. URL: https://www.sciencedirect.com/science/article/pii/S1532046419302436.

[KTC18] Hao-Cheng Kao, Kai-Fu Tang, and Edward Chang. "Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[KVC+21] Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. "Mt-opt: continuous multi-task robotic reinforcement learning at scale". In: *CoRR* abs/2104-.08212 (2021). arXiv: 2104.08212. URL: https://arxiv.org/abs/2104.08212.

[LB94] Henry J. Lowe and G. Octo Barnett. "Understanding and Using the Medical Subject Headings (MeSH) Vocabulary to Perform Literature Searches". In: *JAMA* 271.14 (Apr. 1994), pp. 1103–1108. ISSN: 0098-7484. DOI: 10.1001/jama.1994.03510380059038. URL: https://doi.org/10.1001/jama.1994.03510380059038.

[LBM+24] Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. *Biomistral: a collection of open-source pretrained large language models for medical domains*. 2024. arXiv: 2402.10373 [cs.CL].

[LGB+19] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A. Smith. "Linguistic knowledge and transferability of contextual representations". In: *Proceedings of NAACL 2019*. 2019.

[LHM93] Donald AB Lindberg, Betsy L Humphreys, and Alexa T McCray. "The unified medical language system". In: *Yearbook of medical informatics* 2.01 (1993), pp. 41–51.

[Lip16] Zachary Chase Lipton. "The mythos of model interpretability". In: *ACM Queue* (2016).

[LLG20] Hongyin Luo, Shang-Wen Li, and James Glass. "Knowledge grounded conversational symptom detection with graph memory networks". In: *Proceedings of the 3rd Clinical Natural Language Processing Workshop*. 2020, pp. 136–145.

[LLL+23] Wenzhe Li, Hao Luo, Zichuan Lin, Chongjie Zhang, Zongqing Lu, and Deheng Ye. "A survey on transformers in reinforcement learning". In: *arXiv preprint arXiv:2301.03044* (2023).

[LLN+18] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. "Rllib: abstractions for distributed reinforcement learning". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 3053–3062.

[Llo82] Stuart P. Lloyd. "Least squares quantization in pcm". In: *IEEE Trans. Information Theory* (1982).

[LMJ16] Jiwei Li, Will Monroe, and Dan Jurafsky. "Understanding neural networks through representation erasure". In: *CoRR* (2016).

[LOG+19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "Roberta: A robustly optimized BERT pretraining approach". In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

[LOZ+23] Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. *Llm-qat: data-free quantization aware training for large language models*. 2023. arXiv: 2305.17888 [cs.CL].

[LPP+20] Patrick Lewis et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks". In: *ArXiv* abs/2005.11401 (2020). URL: https://api.semanticscholar.org/CorpusID:218869575.

[LPP+22] Shuang Li et al. “Pre-trained language models for interactive decision-making”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 31199–31212. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/ca3b1f24fc0238edf5ed1ad226b9d655-Paper-Conference.pdf.

[LR02] Xin Li and Dan Roth. “Learning question classifiers”. In: *Proceedings of COLING 2002*. 2002.

[LRH+21] Yuchen Liang, Chaitanya K. Ryali, Benjamin Hoover, Leopold Grinberg, Saket Navlakha, Mohammed J. Zaki, and Dmitry Krotov. “Can a fruit fly learn word embeddings?” In: *CoRR* abs/2101.06887 (2021). arXiv: 2101.06887. URL: https://arxiv.org/abs/2101.06887.

[LYK+20] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. “Biobert: a pre-trained biomedical language representation model for biomedical text mining”. In: *Bioinform.* 36.4 (2020), pp. 1234–1240. DOI: 10.1093/bioinformatics/btz682. URL: https://doi.org/10.1093/bioinformatics/btz682.

[LZZ+20] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. “K-BERT: enabling language representation with knowledge graph”. In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 2901–2908. URL: https://aaai.org/ojs/index.php/AAAI/article/view/5681.

[Maa09] Laurens van der Maaten. “Learning a parametric embedding by preserving local structure”. In: *Proceedings of AISTATS 2009*. 2009.

[MCC+13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space". In: *Workshop Track Proceedings of ICLR 2013*. 2013.

[MKX+18] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. "The natural language decathlon: multitask learning as question answering". In: *CoRR* (2018).

[MLK16] Riccardo Miotto, Li Li, and Brian Kidd. "Deep patient: an unsupervised representation to predict the future of patients from the electronic health records". In: *Scientific Reports* 6 (May 2016), p. 26094. DOI: 10.1038/srep26094.

[MLN19] Paul Michel, Omer Levy, and Graham Neubig. "Are sixteen heads really better than one?" In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett. 2019, pp. 14014–14024. URL: http://papers.nips.cc/paper/9551-are-sixteen-heads-really-better-than-one.

[MR21] Michael Matena and Colin Raffel. "Merging models with fisher-weighted averaging". In: *CoRR* abs/2111.09832 (2021). arXiv: 2111.09832. URL: https://arxiv.org/abs/2111.09832.

[MRK+23] Bhavitvya Malik, Abhinav Ramesh Kashyap, Min-Yen Kan, and Soujanya Poria. "UDAPTER - efficient domain adaptation using adapters". In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Ed. by Andreas Vlachos and Isabelle Augenstein. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 2249–2263. DOI: 10.18653/v1/2023.eacl-main.165. URL: https://aclanthology.org/2023.eacl-main.165.

[MRP+20] Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. "What happens to bert embeddings during fine-tuning?" In: *BlackboxNLP Workshop on Analyzing and Interpret-*

*ing Neural Networks for NLP*. 2020. URL: https://api.semanticscholar.org/CorpusID:216914339.

[NK19] Timothy Niven and Hung-Yu Kao. "Probing neural network comprehension of natural language arguments". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4658–4664. DOI: 10.18653/v1/P19-1459. URL: https://aclanthology.org/P19-1459.

[NKS+19] Galia Nordon, Gideon Koren, Varda Shalev, Eric Horvitz, and Kira Radinsky. "Separating wheat from chaff: joining biomedical knowledge and patient data for repurposing medications". In: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 9565–9572. DOI: 10.1609/aaai.v33i01.33019565. URL: https://doi.org/10.1609/aaai.v33i01.33019565.

[NSM15] Tasha Nagamine, Michael Seltzer, and Nima Mesgarani. "Exploring how deep neural networks form phonemic categories". In: *Proceedings of INTERSPEECH 2015*. 2015.

[OA+23] OpenAI et al. *Gpt-4 technical report*. 2023. arXiv: 2303.08774 [cs.CL].

[OAA+24] OpenAI et al. *Gpt-4 technical report*. 2024. arXiv: 2303.08774 [cs.CL].

[OWJ+22a] Long Ouyang et al. *Training language models to follow instructions with human feedback*. 2022. arXiv: 2203.02155 [cs.CL].

[OWJ+22b] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 27730–27744. URL: https://proceedings.

neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

[PMK24] Pranavi Pathakota, Hardik Meisheri, and Harshad Khadilkar. "Dct: dual channel training of action embeddings for reinforcement learning with large discrete action spaces". In: *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '24. Auckland, New Zealand: International Foundation for Autonomous Agents and Multiagent Systems, 2024, pp. 2411–2413.

[PML21] Paul J. Pritz, Liang Ma, and Kin K. Leung. "Jointly-learned state-action embedding for efficient reinforcement learning". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. CIKM '21. Virtual Event, Queensland, Australia: Association for Computing Machinery, 2021, pp. 1447–1456. ISBN: 9781450384469. DOI: 10.1145/3459637.3482357. URL: https://doi.org/10.1145/3459637.3482357.

[PNI+18] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. "Deep contextualized word representations". In: *Proceedings of NAACL-HLT 2018*. 2018.

[PNI+19] Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. "Knowledge enhanced contextual word representations". 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*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Association for Computational Linguistics, 2019, pp. 43–54. DOI: 10.18653/v1/D19-1005. URL: https://doi.org/10.18653/v1/D19-1005.

[PRR+19] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. "Language models as knowledge bases?" 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*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Association for Computational Linguistics, 2019, pp. 2463–2473. DOI: 10.18653/v1/D19-1250. URL: https://doi.org/10.18653/v1/D19-1250.

[PSR+20] Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. "Stabilizing transformers for reinforcement learning". In: *International conference on machine learning*. PMLR. 2020, pp. 7487–7498.

[PY09] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[QXL+19] Yifan Qiao, Chenyan Xiong, Zheng-Hao Liu, and Zhiyuan Liu. "Understanding the behaviors of BERT in ranking". In: *CoRR* (2019).

[Rad18] Alec Radford. "Improving language understanding by generative pre-training". In: *OpenAI Blog* (2018).

[RBC+21] Jack W. Rae et al. "Scaling language models: methods, analysis & insights from training gopher". In: *CoRR* abs/2112.11446 (2021). arXiv: 2112.11446. URL: https://arxiv.org/abs/2112.11446.

[RBP+18] Salman Razzaki, Adam Baker, Yura Perov, Katherine Middleton, Janie Baxter, Daniel Mullarkey, Davinder Sangar, Michael Taliercio, Mobasher Butt, Azeem Majeed, et al. "A comparative study of artificial intelligence and human doctors for the purpose of triage and diagnosis". In: *arXiv preprint arXiv:1806.10698* (2018).

[RC96] David J Rothwell and RA Cote. "Managing information with snomed: understanding the model." In: *Proceedings of the AMIA Annual Fall Symposium*. American Medical Informatics Association. 1996, p. 80.

[Rei23] Michael V. Reiss. *Testing the reliability of chatgpt for text annotation and classification: a cautionary remark*. 2023. arXiv: 2304.11085 [cs.CL]. URL: https://arxiv.org/abs/2304.11085.

[RHT+17] Maya Rotmensch, Yoni Halpern, Abdulhakim Tlimat, Steven Horng, and David Sontag. "Learning a Health Knowledge Graph from Electronic Medical Records". In: *Scientific Reports* 7.1 (July 2017), p. 5994. ISSN: 2045-2322. DOI: 10.1038/s41598-017-05778-z. URL: https://doi.org/10.1038/s41598-017-05778-z.

[RKR20] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. "A primer in bertology: what we know about how bert works". In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 842–866. URL: https://api.semanticscholar.org/CorpusID:211532403.

[RN18] Alec Radford and Karthik Narasimhan. "Improving language understanding by generative pre-training". In: 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[RT15] Bernardino Romera-Paredes and Philip H. S. Torr. "An embarrassingly simple approach to zero-shot learning". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015, pp. 2152–2161.

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language models are unsupervised multitask learners". In: *OpenAI Blog* (2019).

[RZL+16] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. "Squad: 100, 000+ questions for machine comprehension of text". In: *Proceedings of EMNLP 2016*. 2016.

[SAB+18] Eric W Sayers et al. "Database resources of the National Center for Biotechnology Information". In: *Nucleic Acids Research* 47.D1 (Nov. 2018), pp. D23–D28. ISSN: 0305-1048. DOI: 10.1093/nar/gky1069. eprint: https://academic.oup.com/nar/article-pdf/47/D1/D23/27437595/gky1069.pdf. URL: https://doi.org/10.1093/nar/gky1069.

[SAD+01] Richard N Shiffman, Abha Agrawal, Aniruddha M Deshpande, and Peter Gershkovich. “An approach to guideline implementation with gem”. In: *MEDINFO 2001*. IOS Press. 2001, pp. 271–275.

[SCM+13] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. “Reasoning with neural tensor networks for knowledge base completion”. In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. Ed. by Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger. 2013, pp. 926–934. URL: https://proceedings.neurips.cc/paper/2013/hash/b337e84de8752b27eda3a12363109e80-Abstract.html.

[SDC+19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. “Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *CoRR* abs/1910.01108v4 (2019). arXiv: 1910.01108. URL: http://arxiv.org/abs/1910.01108v4.

[SFA+22] Teven Le Scao et al. “Bloom: a 176b-parameter open-access multilingual language model”. In: *ArXiv* abs/2211.05100 (2022). URL: https://api.semanticscholar.org/CorpusID:253420279.

[SHM+16] David Silver et al. “Mastering the game of go with deep neural networks and tree search”. In: *Nature* 529 (2016), pp. 484–489.

[SHS+17] David Silver et al. “Mastering chess and shogi by self-play with a general reinforcement learning algorithm”. In: *CoRR* abs/1712.01815 (2017). arXiv: 1712.01815. URL: http://arxiv.org/abs/1712.01815.

[SKF+] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. “Bidirectional attention flow for machine comprehension”. In: *Proceedings of ICLR 2017*.

[SLM16] Abigail See, Minh-Thang Luong, and Christopher D. Manning. "Compression of neural machine translation models via pruning". In: *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*. Ed. by Yoav Goldberg and Stefan Riezler. ACL, 2016, pp. 291–301. DOI: 10.18653/v1/k16-1029. URL: https://doi.org/10.18653/v1/k16-1029.

[SM19] Asa Cooper Stickland and Iain Murray. "BERT and pals: projected attention layers for efficient adaptation in multi-task learning". In: *CoRR* abs/1902.02671 (2019). arXiv: 1902.02671. URL: http://arxiv.org/abs/1902.02671.

[SPK16] Xing Shi, Inkit Padhi, and Kevin Knight. "Does string-based neural MT learn source syntax?" In: *Proceedings of EMNLP 2016*. 2016.

[SWD+17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms". In: *CoRR* abs/1707.06347 (2017). arXiv: 1707.06347. URL: http://arxiv.org/abs/1707.06347.

[SZK+22] J Schulman, B Zoph, C Kim, J Hilton, J Menick, J Weng, JFC Uribe, L Fedus, L Metz, M Pokorny, et al. *Chatgpt: optimizing language models for dialogue*. 2022.

[TKC+16] Kai-Fu Tang, Hao-Cheng Kao, Chun-Nan Chou, and Edward Y Chang. "Inquire and diagnose: neural symptom checking ensemble using deep reinforcement learning". In: *NIPS workshop on deep reinforcement learning*. 2016.

[TLI+23] Hugo Touvron et al. *Llama: open and efficient foundation language models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[Top19] Eric Topol. "High-performance medicine: the convergence of human and artificial intelligence". In: *Nature Medicine* 25 (Jan. 2019). DOI: 10.1038/s41591-018-0300-7.

[TXC+19] Ian Tenney et al. "What do you learn from context? probing for sentence structure in contextualized word representations". In: *Proceedings of ICLR 2019*. 2019.

[TZD+20] Eleni Triantafillou et al. “Meta-dataset: a dataset of datasets for learning to learn from few examples”. In: *International Conference on Learning Representations (submission)*. 2020.

[Voo01] Ellen Voorhees. “Overview of trec 2001”. In: *Proceedings of TREC 2001*. 2001.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. “Attention is all you need”. In: *Proceedings of NIPS 2017*. 2017.

[WBC+16] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. “Towards ai-complete question answering: A set of prerequisite toy tasks”. In: *Proceedings of ICLR 2016*. 2016.

[WDS+20] Thomas Wolf et al. “Transformers: state-of-the-art natural language processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[WFL+23] B. Winter, A. Figueroa, A. Löser, F. A. Gers, and Ralf Krestel. ““ddxgym: online transformer policies in a knowledge-graph based natural language environment”. In: *preprint* (2023).

[WGZ+21] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. “KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation”. In: *Transactions of the Association for Computational Linguistics* 9 (Mar. 2021), pp. 176–194. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00360. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00360/1894315/tacl_a_00360.pdf. URL: https://doi.org/10.1162/tacl%5C_a%5C_00360.

[WHM+11] Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. “Ontonotes: a large training corpus for enhanced processing”. In: *Handbook of Natural Language Pro-*

*cessing and Machine Translation: DARPA Global Autonomous Language Exploitation*. Springer, Heidelberg, 2011.

[WIG+22] Mitchell Wortsman et al. *Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time*. 2022. arXiv: 2203.05482 [cs.LG].

[WLK+20] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. *Linformer: self-attention with linear complexity*. 2020. arXiv: 2006.04768 [cs.LG].

[WLP+18] Zhongyu Wei, Qianlong Liu, Baolin Peng, Huaixiao Tou, Ting Chen, Xuanjing Huang, Kam-fai Wong, and Xiangying Dai. "Task-oriented dialogue system for automatic diagnosis". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 201–207. DOI: 10.18653/v1/P18-2033. URL: https://aclanthology.org/P18-2033.

[WRL+22] Benjamin Winter, Alexei Figueroa Rosero, Alexander Löser, Felix Alexander Gers, and Amy Siu. "KIMERA: injecting domain knowledge into vacant transformer heads". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Ed. by Frédéric Calzolari Nicoletta Béchet et al. Marseille, France: European Language Resources Association, June 2022, pp. 363–373. URL: https://aclanthology.org/2022.lrec-1.38.

[WSM+19] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. "GLUE: a multi-task benchmark and analysis platform for natural language understanding". In: (2019). URL: https://openreview.net/forum?id=rJ4km2R5t7.

[WTD+20] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. "K-adapter: infusing knowledge into pre-trained models with adapters". In: *CoRR* abs/2002.01808v5 (2020). arXiv: 2002.01808. URL: https://arxiv.org/abs/2002.01808v5.

[XLS+19a] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. "Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 41.09 (Sept. 2019), pp. 2251–2265. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2018.2857768.

[XLS+19b] Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. "Bert post-training for review reading comprehension and aspect-based sentiment analysis". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. June 2019.

[YCW+19] Zhi-Xiu Ye, Qian Chen, Wen Wang, and Zhen-Hua Ling. "Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models". In: *CoRR* abs/1908.06725v5 (2019). arXiv: 1908.06725. URL: http://arxiv.org/abs/1908.06725v5.

[YML19] Liang Yao, Chengsheng Mao, and Yuan Luo. "KG-BERT: BERT for knowledge graph completion". In: *CoRR* abs/1909-.03193v2 (2019). arXiv: 1909.03193. URL: http://arxiv.org/abs/1909.03193v2.

[YQZ+18] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. "Hotpotqa: A dataset for diverse, explainable multi-hop question answering". In: *Proceedings of EMNLP 2018*. 2018.

[YRH+20] Shunyu Yao, Rohan Rao, Matthew Hausknecht, and Karthik Narasimhan. "Keep calm and explore: language models for action generation in text-based games". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 8736–8754.

[YTC+23] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. *Ties-merging: resolving interference when merging models*. 2023. arXiv: 2306.01708 [cs.LG].

[YY21] Hongyi Yuan and Sheng Yu. "Efficient symptom inquiring and diagnosis via adaptive alignment of reinforcement learning and classification". In: *arXiv preprint arXiv:2112.00733* (2021).

[YYY+24] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. *Language models are super mario: absorbing abilities from homologous models as a free lunch*. 2024. arXiv: 2311.03099 [cs.CL].

[ZAW+19] Ming Zhu, Aman Ahuja, Wei Wei, and Chandan K Reddy. "A hierarchical attention retrieval model for healthcare question answering". In: (2019), pp. 2472–2482.

[ZDW20] Xiao Zhang, Dejing Dou, and Ji Wu. "Learning conceptual-contextual embeddings for medical text". In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 9579–9586. URL: https://aaai.org/ojs/index.php/AAAI/article/view/6504.

[ZMB+14] Xuezhong Zhou, Jörg Menche, Albert-Laszlo Barabasi, and Amitabh Sharma. "Human symptoms–disease network". In: *Nature communications* 5 (June 2014), p. 4212. DOI: 10.1038/ncomms5212.

[ZWM19] Peixiang Zhong, Di Wang, and Chunyan Miao. "Knowledge-enriched transformer for emotion detection in textual conversations". 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*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Association for Computational Linguistics, 2019, pp. 165–176. DOI: 10.18653/v1/D19-1016. URL: https://doi.org/10.18653/v1/D19-1016.

[ZXQ+20] Chen Zhao, Chenyan Xiong, Xin Qian, and Jordan Boyd-Graber. "Complex factoid question answering with a free-text knowledge graph". In: *Proceedings of The Web Conference 2020*. WWW '20. Taipei, Taiwan: Association for Computing Machinery, 2020, pp. 1205–1216. ISBN: 9781450370233. DOI: 10.1145/3366423.3380197. URL: https://doi.org/10.1145/3366423.3380197.

[ZZ18] Quan-shi Zhang and Song-chun Zhu. "Visual interpretability for deep learning: a survey". In: *Frontiers of IT & EE* (2018).