ADVANCING CLINICAL NATURAL LANGUAGE PROCESSING THROUGH KNOWLEDGE-INFUSED LANGUAGE MODELS

by

QIUHAO LU

A DISSERTATION

Presented to the Department of Computer Science
and the Division of Graduate Studies of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

September 2023

DISSERTATION APPROVAL PAGE

Student: Qiuhao Lu

Title: Advancing Clinical Natural Language Processing through Knowledge-Infused Language Models

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Computer Science by:

Thien Huu Nguyen	Chair
Thanh Nguyen	Core Member
Humphrey Shi	Core Member
Margaret E. Sereno	Institutional Representative

and

Krista Chronister	Vice Provost for Graduate Studies

Original approval signatures are on file with the University of Oregon Division of Graduate Studies.

Degree awarded September 2023

DISSERTATION ABSTRACT

Qiuhao Lu

Doctor of Philosophy

Department of Computer Science

September 2023

Title: Advancing Clinical Natural Language Processing through Knowledge-Infused Language Models

Pre-trained Language Models (PLMs) have shown remarkable success in general-domain text tasks, but their application in the clinical domain is constrained by specialized language, terminology, and a lack of in-depth understanding of scientific and medical knowledge. As the adoption of Electronic Health Records (EHRs) and intricate clinical documents continues to grow, the need for domain-adapted PLMs in healthcare research and applications becomes increasingly vital. This research proposes innovative strategies to address these challenges, integrating domain-specific knowledge into PLMs to enhance their efficacy in healthcare. Our approach includes (i) fine-tuning models with knowledge graphs and domain-specific textual data, using graph representation learning and data augmentation techniques, and (ii) directly injecting domain knowledge into PLMs through the use of adapters. By employing these methods, the study aims to improve the performance of clinical language models in tasks such as interpreting EHRs, extracting information from clinical documents, and predicting patient outcomes. The advancements achieved in this work hold the potential to significantly influence the field of clinical Natural Language Processing (NLP) and contribute to improved patient care and healthcare innovation.

This dissertation includes previously published and unpublished co-authored material.

CURRICULUM VITAE

NAME OF AUTHOR: Qiuhao Lu

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, OR, USA
Xi'an Jiaotong University, Xi'an, Shaanxi, China

DEGREES AWARDED:

Master of Science, Control Science and Engineering, 2018, Xi'an Jiaotong University
Bachelor of Science, Internet of Things, 2015, Xi'an Jiaotong University

AREAS OF SPECIAL INTEREST:

Natural Language Processing
Deep Learning
Clinical NLP

PROFESSIONAL EXPERIENCE:

Graduate Research Assistant, University of Oregon, 2018 - 2023
Research Intern - Bioinformatics (PHD) , Mayo Clinic, 2022

GRANTS, AWARDS AND HONORS:

BIBM Student Travel Award, 2021

PUBLICATIONS:

Qiuhao Lu, Dejing Dou, and Thien Huu Nguyen. "ClinicalT5: A Generative Language Model for Clinical Text." *Findings of the Association for Computational Linguistics: EMNLP 2022.*

Qiuhao Lu, Sairam Gurajada, Prithviraj Sen, Lucian Popa, Dejing Dou, and Thien Huu Nguyen. "Cross-lingual Short-text Entity Linking: Generating Features for Neuro-Symbolic Methods." *4th Workshop on Data Science with Human-in-the-loop (DaSH@EMNLP), 2022.*

Qiuhao Lu, Dejing Dou, and Thien Huu Nguyen. "Textual Data Augmentation for Patient Outcomes Prediction." *IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2021.*

Qiuhao Lu, Dejing Dou, and Thien Huu Nguyen. "Parameter-Efficient Domain Knowledge Integration from Multiple Sources for Biomedical Pre-trained Language Models." *Findings of the Association for Computational Linguistics: EMNLP 2021.*

Qiuhao Lu, Thien Huu Nguyen, and Dejing Dou. "Predicting Patient Readmission Risk from Medical Text via Knowledge Graph Enhanced Multiview Graph Convolution." *44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2021.*

Hang Jiang, Sairam Gurajada, **Qiuhao Lu**, Sumit Neelam, Lucian Popa, Prithviraj Sen, Yunyao Li, Alexander Gray. "LNN-EL: A Neuro-Symbolic Approach to Short-text Entity Linking." *ACL 2021.*

Qiuhao Lu, Nisansa de Silva, Dejing Dou, Thien Huu Nguyen, Prithviraj Sen, Berthold Reinwald, and Yunyao Li. "Exploiting Node Content for Multiview Graph Convolutional Network and Adversarial Regularization." *28th International Conference on Computational Linguistics (COLING), 2020.*

Qiuhao Lu, Nisansa de Silva, Sabin Kafle, Jiazhen Cao, Dejing Dou, Thien Huu Nguyen, Prithviraj Sen, Brent Hailpern, Berthold Reinwald, and Yunyao Li. "Learning electronic health records through hyperbolic embedding of medical ontologies." *10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (ACM-BCB), 2019.*

Qiuhao Lu, and Youtian Du. "Wikipedia-based Entity Semantifying in Open Information Extraction." *14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017.*

ACKNOWLEDGEMENTS

I am deeply grateful to have had the opportunity to undertake this incredible journey of earning my Ph.D., and there are many individuals without whom this would not have been possible.

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Thien Nguyen. His patient guidance, thoughtful insights, and consistent support have been instrumental in my growth and development as a researcher. His valuable advice and encouragement have not only shaped my work but also significantly influenced my overall approach to research.

I owe a significant debt of gratitude to Dr. Dejing Dou, who brought me into the program and consistently provided a broad and insightful perspective on my work. His tremendous knowledge and experience, along with his unwavering support, have been fundamental to my accomplishments.

I extend a special thank you to Cheri Smith, whose immense help and assistance have eased my way along this often arduous path. Her assistance, support, and encouragement have played a significant role in enabling me to complete this journey.

To my dear family, I owe an enormous debt of gratitude for their enduring love, unyielding faith in my abilities, and constant motivation throughout this challenging journey. Their constant reassurances and relentless positivity have given me the strength to face all obstacles and see this journey through to the end.

I hope that this dissertation, with all its merits and imperfections, can serve as a fitting testament to the efforts and contributions of all these people. My sincerest thanks to you all.

Dedicated to a silent muse on this journey.

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF FIGURES

LIST OF TABLES

Table Page

CHAPTER I

INTRODUCTION

The majority of the content in this chapter is derived from my dissertation proposal, with me as its principal author. Thien Huu Nguyen contributed by offering invaluable editorial guidance.

1.1 Dissertation Statement

Pre-trained Language Models (PLMs) have emerged as a powerful tool for natural language processing (NLP) in recent years, showing remarkable performance on a wide range of general-domain text tasks. However, they often struggle when applied to domain-specific text due to the problem of domain shift. This is particularly relevant in the clinical domain, where specialized language and terminology are commonly used.

As the use of Electronic Health Records (EHRs) and other clinical data sources becomes more widespread, the need for domain-specific NLP methods has become increasingly apparent. To address this need, a range of domain-specific pre-trained language models has been developed for the clinical domain. These models are trained on large amounts of clinical text data, including EHRs and clinical documents, enabling them to better understand and process technical and specialized language used in the field.

While there have been significant efforts to produce stronger and larger domain-specific pre-trained language models in the clinical domain, most of these models rely on self-supervised pre-training over large amounts of textual data. Recently, ChatGPT has gained attention due to its remarkable performance on various NLP-related tasks across multiple domains, including the biomedical and clinical fields. However, the model is still considered "unhelpful" for medicine

by human experts compared to other domains, indicating a lack of in-depth understanding of domain knowledge. This has led to a growing interest in incorporating external domain-specific knowledge sources into these models to improve their accuracy and efficiency.

In this proposed research work, we aim to innovate in the field of clinical NLP by developing and evaluating novel techniques for knowledge integration into clinical language models and their downstream applications. Our approach to this objective bifurcates into two main strategies:

(i) Fine-tuning the models with knowledge data: We will explore two types of resources for this fine-tuning: knowledge graphs and textual data, leading to two distinct chapters in the dissertation:

   – Harnessing Knowledge Graphs: Integration Techniques for Language Models in Healthcare (Chapter III): Here, we will exploit graph representation learning to enhance the performance of clinical language models and applications.

   – Text-Based Knowledge Infusion: Strategies for Data Augmentation and Beyond (Chapter IV): In this chapter, we will delve into various strategies to augment the training data of clinical language models with external domain-specific knowledge sources and synthetic data.

(ii) Injecting adapters into pre-trained language models: This strategy, presented in Chapter V, titled "Domain Adaptation with Adapters: Parameter-Efficient Approaches to Knowledge Incorporation," aims to directly integrate domain knowledge into pre-trained language models, thereby improving their performance on clinical language tasks.

Through rigorous examination and evaluation of these approaches, our objective is to significantly enhance the efficacy of clinical language models. This improvement is expected to broaden the range of their applications, including but not limited to, efficient interpretation of EHRs, proficient extraction of information from clinical documents, and more accurate prediction of patient outcomes like readmission and mortality rates. These substantial contributions hold the potential to significantly advance the field of clinical NLP, ultimately leading to improved patient outcomes and fostering innovation in healthcare.

1.2 Dissertation Outline

The dissertation unfolds as follows. Chapter II offers an exhaustive review of clinical pre-trained language models (PLMs), assessing their effectiveness and suggesting strategies for their improvement. Chapter III delves into the integration of knowledge graphs into machine learning models in healthcare, with graph representation learning techniques, showcasing three distinct studies that underscore the value of such integration. In Chapter IV, we introduce two innovative text-based data augmentation methods to enhance clinical language models: the generation of synthetic clinical notes and the development of ClinicalT5, a specialized transformer model yielding improved performance in clinical tasks. Lastly, Chapter V investigates a parameter-efficient method for domain knowledge integration into PLMs using adapters, which results in enhanced performance on diverse biomedical NLP tasks without compromising the models' general-domain knowledge.

CHAPTER II

LITERATURE REVIEW: CLINICAL NATURAL LANGUAGE PROCESSING AND PRE-TRAINED LANGUAGE MODELS

This chapter is an adapted version of my area exam, otherwise known as the candidacy exam. As the main author of the initial document, I made substantial contributions, while Thien Huu Nguyen offered indispensable editorial advice.

Pre-trained Language Models (PLMs) have become a crucial tool for natural language processing over the past few years, showcasing remarkable performance on a wide range of applications. However, applying general-domain PLMs to domain-specific text, such as clinical language, has its challenges, leading to the development of domain-specific pre-trained models.

This chapter provides a comprehensive review of the clinical PLMs. We start with a brief overview of foundational concepts of language modeling, including architectures, data sources, training methods, and more. We then examine the current state-of-the-art clinical PLMs and their corresponding methodologies for downstream tasks in the field, including clinical text classification, named entity recognition, relation extraction, and more. Additionally, we discuss the advantages and limitations of each model, as well as their performance compared to general-domain PLMs and other domain-specific models.

Overall, this literature review aims to provide a comprehensive understanding of the current state-of-the-art clinical PLMs and their associated methodologies, as well as the challenges and opportunities for further improving their performance in the future.

2.1 Introduction

Text representation is a crucial task in natural language processing (NLP), forming the basis of nearly all text-related applications Geigle, Mei, and Zhai (2018); P. Liu et al. (2021). Traditionally, to transform the input text into a vector of numerical data, one can represent the words using bag-of-words or tf-idf (term frequency-inverse document frequency) scores Salton (1991); Salton and Buckley (1988) with one-hot encoding. Such methods can suffer from the *curse of dimensionality* problem as the length of vectors usually equals the size of the vocabulary, and decreased efficiency with increasing data size. Moreover, these representations fail to capture the syntactic or semantic information of the text as they only provide a statistical measure of word importance in a corpus. To overcome these issues, researchers propose word embedding techniques, e.g., Word2Vec Mikolov, Chen, Corrado, and Dean (2013), GloVe Pennington, Socher, and Manning (2014) and FastText Bojanowski, Grave, Joulin, and Mikolov (2017), to represent each word in the vocabulary with a fixed embedding vector. With the development of deep learning LeCun, Bengio, and Hinton (2015), researchers use convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to process the text Y. Kim (2014); Lai, Xu, Liu, and Zhao (2015), with the initialization of word vectors from the aforementioned embedding methods Bojanowski et al. (2017); Lu et al. (2020); Mikolov, Chen, et al. (2013); Pennington et al. (2014). This paradigm achieves significant success over a variety of downstream tasks, e.g., named-entity recognition Chiu and Nichols (2016); Sienčnik (2015), text classification Y. Wang, Huang, Zhu, and Zhao (2016), relation classification Zhou et al. (2016) and question answering Xiong, Zhong, and Socher (2017), etc. However, despite their success, word embeddings are limited

in capturing polysemous words, syntactic structures, and semantic roles, hindering their full potential for use in NLP tasks X. Qiu et al. (2020). For instance, the word *apple* has two different meanings in "eat an apple" and "apple computer", but it is only assigned a fixed vector according to the pre-trained word embeddings as they do not consider the contextual information during vectorization, i.e., they are *non-contextualized* or *static* embeddings.

To address the limitations of non-contextualized word embeddings, researchers have turned to the development of *contextualized* representations. With the development and emergence of the transformer architecture Vaswani et al. (2017), considerable efforts have been put into developing transformer-based pre-trained language models Brown et al. (2020); Clark, Luong, Le, and Manning (2020); Devlin, Chang, Lee, and Toutanova (2019); Lan et al. (2019); M. Lewis et al. (2020); Y. Liu et al. (2019); Radford, Narasimhan, Salimans, Sutskever, et al. (2018); Radford et al. (2019); Raffel et al. (2020); Z. Yang et al. (2019). Essentially, the attention mechanism within the transformer allows for more GPU-based parallel computation than Long Short-Term Memory (LSTM) Hochreiter and Schmidhuber (1997), one of the most popular and successful recurrent neural networks for text encoding, and it further facilitates large-scale pre-training and leads to the success of the aforementioned language models. The "pre-train and fine-tune" paradigm has also been a standard approach in modern NLP for a long time. Mascio *et al.* present a comparative analysis on the impact of different text representation methods, i.e., BOW, traditional methods, and BERT Devlin et al. (2019), on selected classification tasks of clinical significance Mascio et al. (2020).

There have been plenty of pre-trained language models over the last few years, e.g., BERT Devlin et al. (2019), GPT-1&2&3 Brown et al. (2020); Radford

et al. (2018, 2019), RoBERTa Y. Liu et al. (2019), ALBERT Lan et al. (2019), T5 Raffel et al. (2020), BART M. Lewis et al. (2020), etc. These models roughly fall into three categories based on their different pre-training frameworks: *decoder*, *encoder*, and *encoder-decoder*. BERT (Bidirectional Encoder Representations from Transformers) drives large-scale self-supervised pre-training on extensive text corpora through the use of Masked Language Modeling (MLM). This involves masking a random subset of tokens in pre-training text and asking the model to predict the original value of the masked tokens. The self-supervised pre-training approach allows the model to learn contextualized text representations from large unannotated text corpora, such as the web, without human supervision H. Wang, Li, Wu, Hovy, and Sun (2022). BERT also introduces Next Sentence Prediction (NSP) which aims to predict whether a given sentence follows the previous sentence or not (i.e., by `[CLS]`). Although NSP is intended to help the model understand longer-term dependencies and relationships across sentences, it is often considered unnecessary and dropped in follow-up works Gu et al. (2021); Joshi et al. (2020); Y. Liu et al. (2019). Unlike BERT, GPT (Generative Pre-trained Transformer) utilizes a decoder-only transformer architecture and performs an autoregressive pre-training task where they seek to predict the next token given existing ones Radford et al. (2018). Moreover, BART (Bidirectional and Autoregressive Transformers) uses an encoder-decoder architecture and employs a denoising sequence-to-sequence pre-training task where the decoder reconstructs the original sentence from a corrupted input, and the model essentially combines bidirectional and autoregressive transformers M. Lewis et al. (2020). Generally, these models differ in their architectures, pre-training objectives, and the data they use. We will delve deeper into these differences in Section 2.2."

In spite of the success of these pre-trained language models on general-domain text, they struggle with domain-specific text due to the problem of *domain shift* Ma, Xu, Wang, Nallapati, and Xiang (2019). As the modern "pre-train and fine-tune" paradigm is a natural fit to domains where large-scaled unannotated textual data is available P. Liu et al. (2021), domain-specific pre-trained language models are been proposed to bridge the gap. In the biomedical and clinical domain, a variety of domain-specific PLMs have been explored and released, including BioBERT Lee et al. (2020), SciBERT Beltagy, Lo, and Cohan (2019), BlueBERT Y. Peng, Yan, and Lu (2019b), ClinicalBERT Huang, Altosaar, and Ranganath (2019), BioClinicalBERT Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019), ClinicalXLNet Huang et al. (2020), umlsBERT Michalopoulos, Wang, Kaka, Chen, and Wong (2020), diseaseBERT Y. He, Zhu, Zhang, Chen, and Caverlee (2020a), ouBioBERT Wada et al. (2020), PubMedBERT Gu et al. (2021), SciFive Phan et al. (2021), BioBART H. Yuan et al. (2022), ClinicalT5 Lu, Dou, and Nguyen (2022), etc.

Besides obtaining domain knowledge via pre-training, another line of research is knowledge infusion where domain knowledge is deliberately injected into language models to enhance their representation capability Y. He, Zhu, Zhang, Chen, and Caverlee (2020b); B. Kim, Hong, Ko, and Seo (2020); Levine et al. (2020); Lu, Dou, and Nguyen (2021a); T. Sun et al. (2020b); X. Wang et al. (2021); Yao, Mao, and Luo (2019b); Z. Zhang et al. (2019). One approach is to incorporate additional knowledge during pre-training. This can be achieved through an auxiliary knowledge-driven training objective. For example, KG-BERT Yao et al. (2019b) integrates factual knowledge from Wikipedia into its model through a knowledge graph completion task, while KEPLER X. Wang et al. (2021)

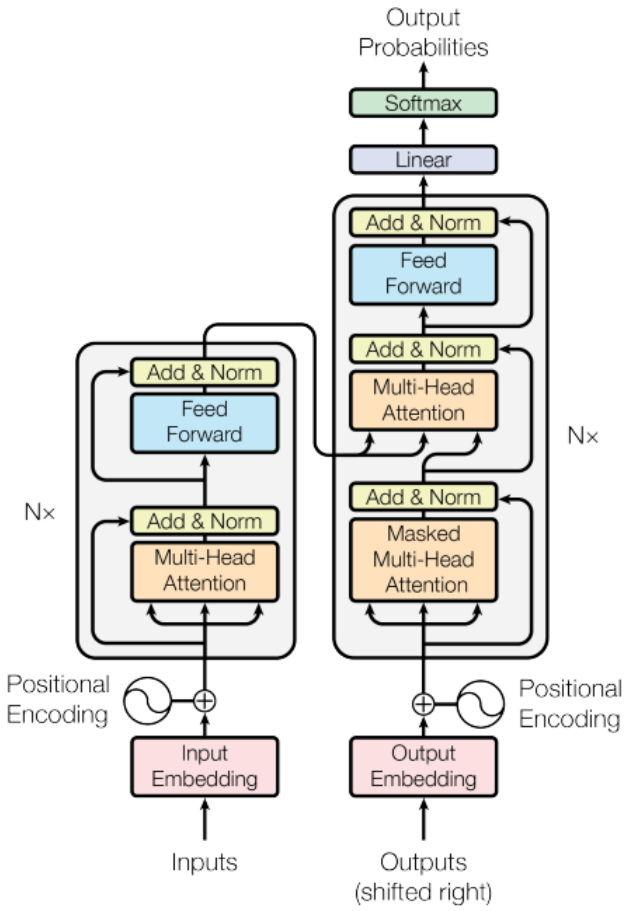


Figure 1. The Transformer model architecture Vaswani et al. (2017).

combines a language modeling objective with a Knowledge Embedding objective for joint optimization. In the clinical domain, there is also some exploration of this direction. For instance, DiseaseBERT seeks to enhance BERT and ALBERT by incorporating disease information through additional pre-training Y. He et al. (2020b). DAKI (Diverse Adapters for Knowledge Integration) incorporates adapters to infuse domain knowledge of multiple sources and formats into PLMs, facilitating the integration of this knowledge in an efficient manner Lu, Dou, and Nguyen (2021a).

It is worth noting that, though the two domains (i.e., biomedical and clinical) are relatively close and the two types of text are similar in many ways, they have some important differences. Clinical text refers to text that is related to the practice of medicine and healthcare service, such as EHRs, physician notes, and other types of text that are commonly used in clinical settings. In contrast, biomedical text refers to text that is related to the field of biomedicine, which includes research articles, textbooks, scientific reports, and other types of text that are used in the study and advancement of biomedicine. In addition, clinical text has unique specific linguistic characteristics, such as the prevalent use of technical jargon, abbreviations, acronyms, passive verbs, and omitted subjects and verbs, which make it distinct from standard language Smith, Megyesi, Velupillai, and Kvist (2014). In this report, we focus on clinical PLMs and will discuss them in Section 2.3.

We also summarize the downstream NLP tasks in the clinical domain, as demonstrated in Section 2.4. For intrinsic tasks, we cover Information Extraction, Text Classification, Semantic Textual Similarity, Question Answering, Question Answering, Text Summarization, Natural Language Inference, etc. For extrinsic tasks, we discuss a bit about patients' outcomes prediction, e.g., readmission, mortality, etc, and clinical predictive tasks, e.g., diagnosis prediction. In the end, we discuss the limitations and potential future directions in Section 2.5.

2.2 Pre-trained Language Models

In this section, we first introduce the key component of modern pre-trained language models, i.e., the transformer architecture Vaswani et al. (2017), and then discuss the most well-known general-domain PLMs in detail, e.g., BERT Devlin et al. (2019), GPT-1&2&3 Brown et al. (2020); Radford et al. (2018, 2019), RoBERTa

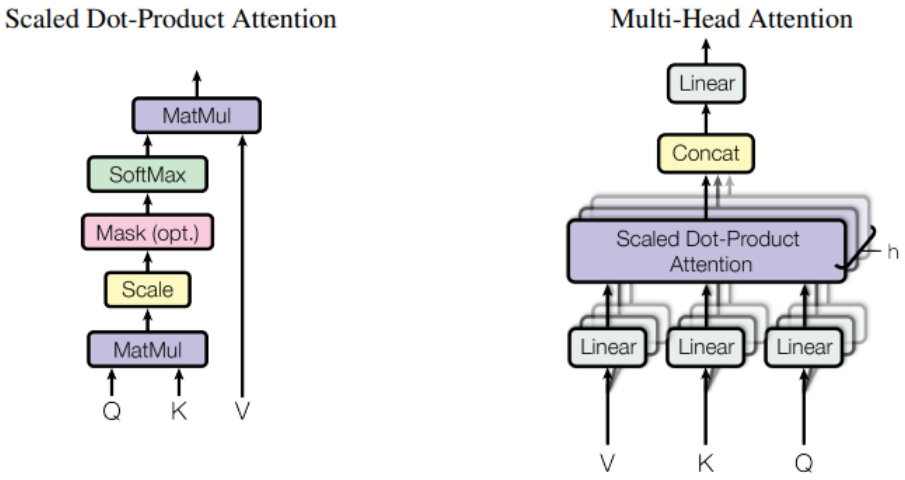


Figure 2. (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel Vaswani et al. (2017).

Y. Liu et al. (2019), ALBERT Lan et al. (2019), T5 Raffel et al. (2020), BART M. Lewis et al. (2020), etc.

2.2.1 Transformer. Recurrent neural networks (RNNs), e.g., long short-term memory networks (LSTM) Hochreiter and Schmidhuber (1997) and gated recurrent neural networks (GRUs), are widely adopted for sequence modeling problems such as language modeling Bengio, Ducharme, and Vincent (2000); Mikolov, Karafiát, Burget, Cernockỳ, and Khudanpur (2010). However, the sequential nature of recurrent models often impedes parallelization within training examples, particularly with longer sequences Vaswani et al. (2017). To overcome this limitation, Vaswani *et al.* introduce the Transformer, a novel transduction model architecture based solely on the *attention* mechanism, eliminating the need for recurrence Vaswani et al. (2017). The transformer architecture allows for significantly more parallel computation and has been one of the key components of large-scale pre-trained language models.

Encoder and Decoder Stacks The architecture of the transformer model is shown in Figure 1. Generally, it consists of an *encoder* and a *decoder*, both of which are stacks of transformer modules. The encoder consists of Nx identical modules and each module has two sublayers, i.e., a multi-head self-attention layer and a position-wise fully connected feed-forward network. Within each sublayer, there is also a residual connection K. He, Zhang, Ren, and Sun (2016) and a layer normalization operation Ba, Kiros, and Hinton (2016) that are leveraged to improve the performance and training efficiency (i.e., Add&Norm). The decoder has a similar architecture to the encoder, except for an additional multi-head attention sublayer over the output of the encoder. The self-attention sublayer in the decoder is a bit different from that in the encoder, where future values are masked out to avoid information leakage and preserve the autoregressive property.

Attention The attention mechanism is a core component in many deep learning models, especially in the field of natural language processing. It allows a model to focus its attention on specific parts of an input, such as words or phrases in a sentence when making predictions. The attention mechanism works by computing a weight for each element of the input and then using these weights to calculate a weighted sum of the elements as the output. The weights are determined by a compatibility function that measures the similarity between a query vector and key vectors associated with each element. In the Transformer architecture, attention is implemented using a combination of linear transformations and softmax activation functions. Unlike recurrent neural networks (RNNs), which use sequential computations, the linear transformations used in the Transformer's attention mechanism are relatively simple, allowing for efficient parallel computation.

In particular, self-attention is a mechanism used in deep learning models to capture dependencies between elements in a sequence of inputs. Essentially, it represents each input token as a weighted sum of all the token vectors in the input where the weights are computed based on the relationships between them.

In the transformer, the self-attention is implemented as "Scaled Dot-Product Attention" as shown in Figure 2. Generally, they compute the dot products of the query Q with all keys K and divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values V:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

Multi-head attention is a mechanism that allows a model to attend to multiple, different parts of the input sequence at once, instead of focusing on just one part as in single-head attention where the meaning of a word may largely depend on itself Kalyan, Rajasekharan, and Sangeetha (2021). In multi-head attention, the input sequence is transformed into multiple separate, parallel representations, each of which is passed through a separate attention mechanism, i.e., attention is applied multiple times in parallel. Consequently, this mechanism allows the model to capture multiple types of relationships between elements in the sequence.

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_\text{h})W^O \\ \text{where head}_\text{i} &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.2}$$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$,$W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$,$W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

The Transformer uses multi-head attention in three different ways. The first type is the self-attention layer in the *encoder* where each position attends to all

the words in the input sequence. The second type is the self-attention layer in the *decoder*. Similarly, each position attends to all positions up to that position where the future values are masked out (set to $-\infty$), i.e., masked self-attention. The third type is cross-attention within the *encoder-decoder* architecture where each position in the decoder attends to all positions in the input sequence.

Position-wise Feed-Forward Networks In addition to the multi-head attention mechanism, each encoder and decoder in the Transformer architecture also includes a feed-forward neural network, as depicted in Figure 1. The feed-forward network operates in a position-independent manner, applying the same linear transformation to each element in the sequence using identical parameters. The parameters are not shared across different layers.

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b2 \tag{2.3}$$

Positional Encoding As there are no recurrent neural networks (RNNs) that are supposed to preserve the positional information of the input sequence in the transformer, the architecture incorporates the technique *Positional Encoding* Gehring, Auli, Grangier, Yarats, and Dauphin (2017) that injects a position embedding vector into individual input embeddings. This is achieved by adding a position-specific embedding vector to the embedded representation of each word. These position embedding vectors follow a learned periodic function that allows the model to determine the relative position of each word in the sequence.

2.2.2 Methods of PLMs.

There has been a surge of interest in developing different pre-trained language models in the past few years, e.g., BERT Devlin et al. (2019), GPT-1&2&3 Brown et al. (2020); Radford et al. (2018, 2019), RoBERTa Y. Liu et al. (2019), ALBERT Lan et al. (2019), T5 Raffel et

Model	Framework	Pre-training Method
BERT Devlin et al. (2019)	Encoder	MLM, NSP
RoBERTa Y. Liu et al. (2019)	Encoder	MLM
ALBERT Lan et al. (2019)	Encoder	MLM, SOP
XLM-R Conneau et al. (2020)	Encoder	MLM
ELECTRA Clark et al. (2020)	Encoder	RTD
XLNet Z. Yang et al. (2019)	Decoder	PLM
GPT Radford et al. (2018)	Decoder	CLM
T5 Raffel et al. (2020)	Encoder-Decoder	Seq2seq MLM

Table 1. Representative general-domain PLMs. Underexplored models in the clinical scenario are omitted for simplicity.

al. (2020), BART M. Lewis et al. (2020), etc. These models can be classified into three categories based on their pre-training frameworks: *decoder*, *encoder*, and *encoder-decoder*. In this subsection, we introduce some of the prevalent pre-training frameworks that lay the foundations of clinical PLMs and discuss their corresponding applications.

Decoder-only models (or autoregressive models) refer to models pre-trained based on the language modeling task, i.e., predicting the next token given observed ones, which also corresponds to the decoder of the transformer model. As mentioned above, GPT is a typical decoder-only pre-trained language model Radford et al. (2018). Essentially, GPT computes the probability distribution of the next token given previous tokens, with the decoder module of the original transformer, for pre-training. The model is pre-trained on the Book Corpus dataset and demonstrates new SOTA results on several NLP benchmarks Radford et al. (2018). GPT-2 Radford et al. (2019) and GPT-3 Brown et al. (2020) are the 2nd and 3rd release of GPT, which generally share the same architecture with the original version, i.e., the transformer decoder, and have 1.5 billion and 175

billion model parameters, respectively. Both GPT-2 and GPT-3 can be applied to downstream tasks without fine-tuning, demonstrating the potential of large PLMs with updated SOTA performance.

Encoder-only models (or autoencoding models) refer to models pre-trained based on the reconstruction objective of corrupted input sentences, which also corresponds to the encoder of the transformer model. Besides BERT which depends on Masked Language Modeling and Next Sentence Prediction as mentioned above, RoBERTa is another typical example of such type Y. Liu et al. (2019). Essentially, RoBERTa tackles some of BERT's issues and proposes the dynamic masking technique where they seek to randomly generate the mask at each epoch, as opposed to BERT's static masking strategy. RoBERTa also drops the NSP pre-training task due to its lack of impact and instead puts two consecutive full sentences together as input without asking the model to predict their consecutiveness. ALBERT Lan et al. (2019) generally follows BERT, and it also proposes some useful tricks. Essentially, ALBERT is a light and efficient variant of BERT that differs in three aspects: factorized embedding parameterization, cross-layer parameter sharing and NSP replaced by sentence ordering prediction. Empirically, the performance is better than BERT on a variety of tasks in many aspects. ELECTRA is another pre-training framework for BERT whose key innovation is Replaced Token Detection (RTD, as a replacement for MLM). The task is to simultaneously optimize a generator-discriminator architecture where the generator is trained using the MLM objective given a randomly masked sentence as input, and the discriminator (ELECTRA) aims to predict whether each token is original or generated Clark et al. (2020). ELECTRA demonstrates efficiency and better performance than BERT/RoBERTa across multiple benchmarks.

Encoder-decoder models refer to models pre-trained based on a sequence-to-sequence objective, which also corresponds to the encoder-decoder architecture of the original transformer. As a typical example, BART M. Lewis et al. (2020) takes as input to the encoder a corrupted text with an arbitrary noising function (Token Masking, Token Deletion, Text Infilling, Sentence Permutation, Document Rotation), and the decoder is enforced to reconstruct the original text. The model can be viewed as a combination of a bidirectional encoder (e.g., BERT) and an autoregressive decoder (e.g., GPT), and this architecture makes it better at generative tasks while keeping the bidirectional encoding capabilities. Another example is T5 Raffel et al. (2020) which casts different NLP tasks as a text-to-text problem by assigning a specific prefix. T5 has self-supervised and supervised training. For the self-supervised pre-training, T5 takes a corrupted sentence as input and the self-supervised pre-training task is to generate the dropped-out tokens. The supervised pre-training tasks are transformed downstream tasks from the GLUE and SuperGLUE benchmarks.

Generally, there have been numerous studies on PLMs in the past few years. Some of them are CTRL Keskar, McCann, Varshney, Xiong, and Socher (2019), Transformer-XL Z. Dai et al. (2019), Reformer Kitaev, Kaiser, and Levskaya (2020), XLNet Z. Yang et al. (2019), DistilBERT Sanh, Debut, Chaumond, and Wolf (2019), ConvBERT Z.-H. Jiang et al. (2020), Funnel Transformer Z. Dai, Lai, Yang, and Le (2020), Longformer Beltagy, Peters, and Cohan (2020), ProphetNet Qi et al. (2020), Switch Transformer Fedus, Zoph, and Shazeer (2021), GLaM Du et al. (2022), Gropher Rae et al. (2021), some multi-lingual models like mT5 L. Xue et al. (2021), ERNIE Y. Sun et al. (2021), and so forth. As these models are rarely adopted in the clinical domain, they are not covered in this report.

Model	Type	Initialization	EHR
BEHRT Y. Li et al. (2020)	patient visits (code)	scratch	CPRD
Med-BERT Rasmy, Xiang, Xie, Tao, and Zhi (2021)	patient visits (code)	scratch	Cerner, Truven
BRLTM Meng, Speier, Ong, and Arnold (2021)	patient visits (code)	scratch	private
G-BERT Shang, Ma, Xiao, and Sun (2019)	patient visits (code)	scratch	MIMIC-III

Table 2. Summary of EHR-based clinical PLMs.

2.3 Clinical PLMs

The rapid increase in Electronic Health Records (EHRs) and the wealth of digitized longitudinal clinical data they contain have sparked significant interest in using machine learning techniques to tackle medical challenges Wen et al. (2019). In response to this trend, various domain-specific pre-trained language models have been developed for the clinical domain, in addition to the already existing general-domain models. In this section, we will provide a brief overview of the motivation behind developing and utilizing domain-specific pre-trained language models in the clinical field and then delve into a more in-depth examination of the different clinical PLMs available.

2.3.1 Motivation. In the clinical domain, the reasons for developing and utilizing domain-specific pre-trained language models are straightforward.

In general, the use of domain-specific PLMs in the clinical field is motivated by the need for improved accuracy and efficiency in language-based tasks. Training on large amounts of textual data specific to the clinical domain, such as electronic health records (EHRs) and clinical documents, enables these models to better understand and process the technical and specialized language commonly used in this field, including medical terminology and abbreviations. This can be useful for tasks such as the interpretation of EHRs, extraction of relevant information from clinical documents, generation of clinical summary reports, etc.

2.3.2 Data Resources.

A variety of unannotated and free textual resources are used in pre-training a clinical PLM, such as clinical notes in EHRs, relevant social media posts, scientific literature, external knowledge bases, etc. We refer the readers to Gonzalez-Hernandez, Sarker, O'Connor, and Savova (2017); Kalyan and Sangeetha (2020) for a more detailed treatment of biomedical and clinical textual corpora.

Moreover, as most domain PLMs in the biomedical and clinical domains are variants of BERT, the biggest difference among them is their pre-training data. As a result, we will cover these models, especially the less popular ones, in this subsection.

Electronic Health Records Electronic Health Records have been widely adopted by healthcare providers to electronically record patients' visits and health information in the last few years Henry, Pylypchuk, Searcy, Patel, et al. (2016). There are several reasons why clinical pre-trained language models are often trained on electronic health records (EHRs). First, EHRs contain a wealth of information about patients' health histories and treatment plans, which can be valuable for language models to learn from. This information can include demographics, diagnoses, medications, laboratory test results, radiology images, and more. Second, EHRs are widely used in the healthcare industry, so clinical language models trained on EHRs may be more applicable and useful in real-world settings. Finally, since many healthcare providers use EHR systems, it is often possible to access large amounts of data from these systems for research purposes, although certain privacy and ethical issues must be considered.

The MIMIC-III Critical Care (Medical Information Mart for Intensive Care III) Database is a large, freely-available database composed of de-identified EHR

data Johnson et al. (2016) and has been widely used for clinical NLP research Feng et al. (2022); Lu, Nguyen, and Dou (2021); Rajkomar et al. (2018); Shorten, Khoshgoftaar, and Furht (2021). It is also one of the most popular EHR datasets that are used to train clinical language models, which consists of the EHRs of patients in the intensive care unit (ICU) of the Beth Israel Deaconess Medical Center between 2001 and 2012.

ClinicalBERT[1] is one of the most popular domain variants which initializes from BioBERT Lee et al. (2020) and is further pre-trained on MIMIC-III clinical notes Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019). Another ClinicalBERT has similar settings Huang et al. (2019), where the authors also propose ClinicalXLNet Huang et al. (2020), an XLNet Z. Yang et al. (2019) variant that is further pre-trained on MIMIC-III clinical notes. Similarly, Yang *et al.* propose BERT-MIMIC, ELECTRA-MIMIC, XLNET-MIMIC, RoBERTa-MIMIC, DeBERTa-MIMIC, Longformer-MIMIC based on further pre-training on MIMIC text X. Yang, Bian, Hogan, and Wu (2020). ClinicalT5 further pre-trains SciFive Phan et al. (2021) on MIMIC notes and produces a clinical variant of T5 Lu, Dou, and Nguyen (2022). In general, such models mostly depend on further pre-training on unstructured clinical notes in MIMIC-III. In fact, the MIMIC database consists not only of unstructured textual data but also structured information, including different kinds of numerical features of patients, disease and procedure codes, demographics, etc.

BEHRT Y. Li et al. (2020) is a language model trained from scratch using EHRs, with MLM as the pre-training task. The authors use code, position, age, and segment embeddings to improve the model's performance. Med-BERT Rasmy

[1]Also known as BioClinicalBERT.

et al. (2021) is another language model trained from scratch with MLM and LOS (Length of Stay) as pre-training tasks. The authors use code, serialization, and visit embeddings to further improve the model's ability to handle medical data. BRLTM Meng et al. (2021) is trained from scratch using multi-modal data with MLM. MedGPT Kraljevic et al. (2021) is a GPT-like language model trained on patients' medical histories in the format of EHRs. Given a sequence of past medical events, MedGPT aims to predict future events.

Scientific literature Some clinical pre-trained language models are trained on scientific publications, such as research articles and medical journals because these texts can provide valuable information about current medical knowledge and practices. Scientific publications often contain detailed descriptions of medical conditions, treatments, and research findings, which can be useful for language models to learn from. Training a language model on scientific publications can also help the model to understand medical terminology and concepts more accurately and in greater depth. This can be particularly useful for tasks that involve analyzing or summarizing medical information. Finally, scientific publications may be easier to obtain than other types of clinical data, such as electronic health records (EHRs). Many scientific publications are freely available online, making it possible to create large datasets for training language models.

PubMed is a free online database that provides access to millions of scientific articles and abstracts related to medicine, biology, and life sciences. PubMed Central (PMC) is an open-access digital archive of scientific articles that contains full-text articles in the biomedical and life sciences, making it a valuable resource for researchers. PubMed abstracts (PubMed) and PubMed Central (PMC)

are widely adopted for training language models in the biomedical field. B. Wang et al. (2021).

BioBERT is the first biomedical pre-trained language model which is obtained by further pre-training general BERT on biomedical literature Lee et al. (2020). Similarly, BioMedBERT is obtained by further pretraining BERT-large on the BREATHE dataset Chakraborty et al. (2020). BlueBERT further pre-trains on the PubMed text and de-identified clinical notes from MIMIC-III Y. Peng et al. (2019b), so as BioALBERT Naseem, Dunn, Khushi, and Kim (2022). BioMed-RoBERTa Gururangan et al. (2020) is obtained by further pre-training on 2.68 million full-text papers from S2ORC Lo, Wang, Neumann, Kinney, and Weld (2020), a large corpus of academic papers spanning many academic disciplines including the biomedical domain. Unlike these models, SciBERT builds its own vocabulary and pre-trains from scratch on scientific papers from Semantic Scholar, in which 82% are from the biomedical domain and 18% are from the computer science domain Beltagy et al. (2019). PubMedBERT is obtained by domain-specific pre-training from scratch on PubMed text Gu et al. (2021).

Social media Clinical pre-trained language models may also be trained on social media posts, such as those from *Reddit*, *Twitter*, *AskAPatient*, *WebMD*, in order to learn about common language usage and slang in the context of healthcare. These platforms can provide a large amount of real-world data that can be used to train language models to understand how people discuss healthcare-related topics in everyday language. Training on social media posts can also provide the model with a better understanding of the context which could benefit sentiment or opinion-related tasks. However, it is important to ensure the representativeness and suitability of the data before using it for model training.

Reddit and Twitter are commonly used social media sources for training language models. Reddit is a social media platform that allows users to share news, images, and links, as well as participate in forums and discussions on a wide range of topics. Reddit is considered a valuable resource for language model training because it provides a large and diverse dataset of written content, ranging from informal conversations to in-depth discussions on a wide range of topics. Twitter is a microblogging platform that allows users to post short messages, images, and videos. Similar to Reddit, Twitter also provides a vast amount of textual data, which can help models learn to understand conversational text.

For example, BERTweet Nguyen, Vu, and Tuan Nguyen (2020) is obtained by training on Twitter posts. COVID-twitter-BERT Müller, Salathé, and Kummervold (2020) is a natural language model to analyze COVID-19 content on Twitter. The COVID-twitter-BERT model is initialized from BERTweet and trained on tweets about COVID-19. BioRedditBERT Basaldella, Liu, Shareghi, and Collier (2020) is initialized from BioBERT and further pre-trained on health-related Reddit posts.

External knowledge bases External medical knowledge bases can be complementary to clinical pre-trained language models, as they are often not fully exposed to structured domain knowledge, which may not be sufficiently encoded in the pre-training text. The external knowledge bases often serve more as an auxiliary training objective that works along with typical self-supervised pre-training on large amounts of textual data.

One of the most important knowledge resources is the Unified Medical Language System (UMLS)[2], which is a comprehensive and standardized terminology repository that is widely used in the field of biomedical research and healthcare Bodenreider (2004). It includes a wide range of medical and health-related vocabularies and terminologies, such as NCBI, MeSH, SNOMED CT, ICD-10, Gene Ontology, OMIM, and many others. The UMLS is designed to help researchers, clinicians, and other healthcare professionals communicate effectively and accurately by providing a common language and set of terms that can be used across different systems and contexts. It is maintained and updated by the National Library of Medicine (NLM) in the United States, and all vocabularies are freely available for research purposes under a corresponding license agreement.

For example, Hao *et al.* propose to enhance clinical BERT embedding using a joint further pre-training strategy, where they incorporate a joint loss of masked language modeling, next sentence prediction, and triplet classification on MIMIC-III notes and UMLS relations to obtain Clinical KB-BERT and Clinical KB-ALBERT Hao, Zhu, and Paschalidis (2020). UmlsBERT further pre-trains ClinicalBERT Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019) on MIMIC-III notes with a specifically designed multi-label loss that incorporates UMLS information Michalopoulos et al. (2020). SapBERT further pre-trains PubMedBERT Gu et al. (2021) on UMLS synonyms under a scalable metric learning framework F. Liu, Shareghi, Meng, Basaldella, and Collier (2021). KeBioLM incorporates UMLS entity information by linking PubMed abstracts to the knowledge base and adopts an entity detection/linking objective Z. Yuan, Liu, Tan, Huang, and Huang (2021). Coder injects medical knowledge from UMLS

[2]http://umlsks.nlm.nih.gov

Model	Type	Initialization	Corpora	Publicly Available
ClinicalBERT Huang et al. (2019)	clinical notes	BERT	MIMIC-III	Y
ClinicalBERT Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019)	clinical notes	BioBERT	MIMIC-III	Y
UmlsBERT Michalopoulos et al. (2020)	clinical notes, KG	ClinicalBERT	MIMIC-III, UMLS	Y
DiseaseBERT Y. He et al. (2020a)	Wiki articles	BERT	Wikipedia	Y
PubMedBERT Gu et al. (2021)	scientific literature	scratch	PubMed, PMC	Y
BERT-MIMIC X. Yang et al. (2020)	clinical notes	BERT	MIMIC-III	Y
ELECTRA-MIMIC X. Yang et al. (2020)	clinical notes	ELECTRA	MIMIC-III	Y
XLNET-MIMIC X. Yang et al. (2020)	clinical notes	XLNet	MIMIC-III	Y
RoBERTa-MIMIC X. Yang et al. (2020)	clinical notes	RoBERTa	MIMIC-III	Y
DeBERTa-MIMIC X. Yang et al. (2020)	clinical notes	DeBERTa	MIMIC-III	Y
Longformer-MIMIC X. Yang et al. (2020)	clinical notes	Longformer	MIMIC-III	Y
ClinicalXLNet Huang et al. (2020)	clinical notes	XLNet	MIMIC-III	Y
DiseaseALBERT Y. He et al. (2020a)	Wiki articles	ALBERT	Wikipedia	Y
BioMedBERT Chakraborty et al. (2020)	scientific literature	BERT	BREATHE	N
BlueBERT Y. Peng et al. (2019b)	scientific literature, clinical notes	BERT	PubMed, MIMIC-III	Y
SciBERT Beltagy et al. (2019)	scientific literature	scratch	Semantic Scholar	Y
MedGPT Kraljevic et al. (2021)	clinical notes	GPT	KCH, MIMIC-III	Y
BioMed-RoBERTa Gururangan et al. (2020)	scientific literature	RoBERTa	S2ORC	Y
COVID-twitter-BERT Müller et al. (2020)	social media posts	BERTweet	Twitter	Y
BioRedditBERT Basaldella et al. (2020)	social media posts	BioBERT	Reddit	Y
SapBERT F. Liu et al. (2021)	KG	PubMedBERT	UMLS synonyms	Y
CODER Z. Yuan et al. (2022)	KG	BioBERT	UMLS	Y
KeBioLM Z. Yuan et al. (2021)	KG	PubMedBERT	UMLS	Y
Clinical KB-BERT Hao et al. (2020)	KG	BioBERT	UMLS	Y
Clinical KB-ALBERT Hao et al. (2020)	KG	ALBERT	UMLS	Y
SciFive Phan et al. (2021)	scientific literature	T5	PubMed, PMC	Y
BioALBERT Naseem et al. (2022)	scientific literature	ALBERT	PubMed, PMC	Y
EhrBERT F. Li et al. (2019)	clinical notes	BioBERT	private	N
RoBERTa-PubMed-MIMIC P. Lewis, Ott, Du, and Stoyanov (2020)	scientific literature, clinical notes	RoBERTa	PubMed, PMC, MIMIC-III	Y
GatorTron X. Yang et al. (2022)	scientific literature, clinical notes, articles	scratch	UF Health, PubMed, Wikipedia	Y
UCSF-BERT Sushil, Ludwig, Butte, and Rudrapatna (2022)	clinical notes	scratch	UCSF Health	N
CLIN-X-en Lange, Adel, Strötgen, and Klakow (2022)	clinical PubMed abstracts	XLM-R	PubMed	Y
CLIN-X-es Lange et al. (2022)	clinical notes	XLM-R	Scielo archive, MeSpEn	Y
MedGTX S. Park, Bae, Kim, Kim, and Choi (2022)	EHR	BERT	MIMIC-III	Y
Clinical-Longformer Y. Li, Wehbe, Ahmad, Wang, and Luo (2022)	clinical notes	Longformer	MIMIC-III	Y
Clinical-BigBird Y. Li et al. (2022)	clinical notes	BigBird	MIMIC-III	Y
BioMedLM[3]	scientific literature	GPT	PubMed, PMC	Y
DRAGON Yasunaga, Bosselut, et al. (2022)	scientific literature, KG	BioLinkBERT	PubMed, UMLS	Y
Med-PaLM Singhal et al. (2022)	instructions and exemplars	Flan-PaLM	MultiMedQA, human input	N
ClinicalT5 Lu, Dou, and Nguyen (2022)	clinical notes	SciFive	MIMIC-III	Y
DAKI-BERT Lu, Dou, and Nguyen (2021a)	Wiki articles, KG	BERT	Wikipedia, UMLS	Y
DAKI-ALBERT Lu, Dou, and Nguyen (2021a)	Wiki articles, KG	ALBERT	Wikipedia, UMLS	Y
DAKI-ClinicalBERT Lu, Dou, and Nguyen (2021a)	Wiki articles, KG	ClinicalBERT	Wikipedia, UMLS	Y

KG = knowledge graph

Table 3. Summary of Clinical PLMs.

into BioBERT Lee et al. (2020) through contrastive further training Z. Yuan et al. (2022). DiseaseBERT and DiseaseALBERT are obtained by further pre-training on disease-related articles from Wikipedia Y. He et al. (2020a).

2.3.3 Pre-training Strategies. According to a recent survey on biomedical pre-trained language models, Kalyan *et al.* point out that existing biomedical PLMs roughly fall into the following two categories, i.e., mixed-domain pre-training, and domain-specific pre-training Kalyan et al. (2021).

The situation in the clinical domain is quite similar. In fact, most of the aforementioned clinical/biomedical domain-specific pre-trained language models are based on the mixed-domain pre-training strategy (or continual pre-training), as pre-

training on large amounts of general-domain text is proven beneficial. Essentially, the mixed-domain pre-training strategy initializes with a pre-trained model and continues the pre-training process with domain-specific data and objectives. For example, BioBERT Lee et al. (2020) initializes from BERT Devlin et al. (2019), ClinicalBERT Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019) initializes from BioBERT Lee et al. (2020), ClinicalT5 Lu, Dou, and Nguyen (2022) initializes from SciFive Phan et al. (2021), etc. This strategy demonstrates the issue of inconsistent vocabularies, which results in less representative capability of continual pre-trained models in the target domain Gu et al. (2021). However, existing PLMs mostly use subword tokenization algorithms which effectively alleviate the issue by decomposing rare words into meaningful subwords, such as Byte-Pair Encoding (BPE) Sennrich, Haddow, and Birch (2016), WordPiece Schuster and Nakajima (2012), Unigram Kudo (2018), SentencePiece Kudo and Richardson (2018), etc.

It is important to point out that the mixed-domain pre-training approach is particularly useful when the target domain has a limited amount of text and can benefit from being pre-trained using general-domain text like Wikipedia and BookCorpus Devlin et al. (2019) as well as related-domain text. However, this is not the case for the biomedical domain, as it has a large and growing corpus of text, with over 30 million texts in PubMed and this motivates PubMedBERT which is trained from scratch Gu et al. (2021). Conversely, the clinical domain presents a different scenario. Due to the sensitive nature of the clinical text, such as clinical notes in EHRs, and the difficulties in obtaining such data, most clinical pre-trained language models rely on mixed-domain pre-training, such as ClinicalBERT Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019), ClinicalBERT Huang et al. (2019), SciFive Phan et al. (2021), ClinicalT5 Lu, Dou, and Nguyen (2022), etc.

There are also variants that are trained from scratch, such as PubMedBERT which is trained from scratch on PubMed abstracts and PMC full-text articles Gu et al. (2021), and SciBERT which is trained from scratch on scientific papers from Semantic Scholar Beltagy et al. (2019). Essentially, the domain-specific pre-training (training from scratch) method aims to fix the vocabulary inconsistency issue between the general domain and the biomedical domain Kalyan et al. (2021). It is also worth noting that EHR-based language models are generally pre-trained from scratch such as BEHRT Y. Li et al. (2020), Med-BERT Rasmy et al. (2021), BRLTM Meng et al. (2021), etc., as they depend on code, demographics, visits, etc. instead of clinical narratives.

In order to gain a deeper understanding and provide a comprehensive overview of the training objectives of clinical pre-trained language models, this subsection will explore the various pre-training strategies in detail. It is worth noting that most existing clinical PLMs rely on continual pre-training, which means they would typically use similar pre-training tasks as general-domain models such as BERT Devlin et al. (2019) but fine-tune on a large corpus of clinical data. This is done to capture the specific language and structure of the clinical domain, and improve the models' performance on downstream tasks such as named entity recognition, relation extraction, and de-identification.

In this subsection, we would cover some of the most popular pre-training tasks as well as those adopted in the aforementioned clinical PLMs.

Masked Language Modeling (MLM) This is a task where a random subset of the tokens in a sentence are replaced with a `[MASK]` token and the model is trained to predict the original token based on the context provided by the observed tokens in the sentence. Many models such as BERT Devlin et al. (2019), BioBERT

Lee et al. (2020) and ClinicalBERT Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019) use this pre-training task. As arguably one of the most popular and well-explored pre-training techniques, researchers have proposed several tricks to improve its performance. For example, instead of token masking, Cui *et al.* propose whole word masking for Chinese BERT which demonstrates better performance Cui, Che, Liu, Qin, and Yang (2021). Besides, RoBERTa uses dynamic masking to replace BERT's static masking, where they randomly generate the mask at each epoch Y. Liu et al. (2019) and this trick is also applied in their domain variants, e.g., BioMed-RoBERTa Gururangan et al. (2020). ERNIE incorporates entity-level masking and phrase-level masking which is beneficial to infuse entity knowledge into the model Z. Zhang et al. (2019).

Next Sentence Prediction (NSP) This task involves training the model to predict whether two sentences are contiguous or not. The objective is to learn the sentence-level context in the corpus and it's used by most of the pre-trained models derived by BERT Devlin et al. (2019). Although NSP is intended to help the model understand longer-term dependencies and relationships across sentences, its real impact on the model has been questioned in several studies Gu et al. (2021); Joshi et al. (2020); Y. Liu et al. (2019), as mentioned above.

Replaced Token Detection (RTD) This is a pre-training task that is leveraged to improve robustness to word replacement and text-to-text transfer. In this task, words in a sentence are replaced with other words that have a similar meaning, and the model is trained to detect which words have been replaced. This task helps the model learn to understand the meaning of words and their relationships to other words in a sentence. One example of a model that uses RTD

for pre-training is ELECTRA Clark et al. (2020). The model uses RTD to generate masked tokens and then trains a generator model to predict the original tokens based on the context. The generator is then fine-tuned on a downstream task and the encoder is used for the final classification. The main idea behind ELECTRA is to make the pre-training task more challenging and to reduce the risk of overfitting, by replacing some of the tokens with fake ones. It is worth noting that this task is not being widely used in the clinical domain yet. Some biomedical domain variants of ELECTRA that depend on continual pre-training naturally inherit this method, such as Bio-ELECTRA Ozyurt (2020), BioELECTRA Kanakarajan, Kundumani, and Sankarasubbu (2021), etc.

Sentence Order Prediction (SOP) This task aims to make the model predict the correct order of a set of sentences. Essentially, the key idea is to use two consecutive sentences from the same document as a positive sample, and to swap the two consecutive sentences to make a negative sample. This task helps the model to understand the sequential nature of language and the relationships between sentences in a document. It is worth noting that this task is motivated by the fact that NSP is often dropped by researchers due to its ineffectiveness, as mentioned above. As a replacement, ALBERT proposes SOP based on their conjecture that NSP is not very effective because it mixes both topic prediction and coherence prediction, the former of which is comparatively easy to handle which hinders the optimization of the other task Lan et al. (2019). SOP is applied in domain variants of ALBERT, such as Clinical KB-ALBERT Hao et al. (2020), DiseaseALBERT Y. He et al. (2020a), etc.

Permutation Language Modeling (PLM) This pre-training task aims to train the model to predict the correct order of a sentence given the context provided by the rest tokens of the sentence. Essentially, the input sentence is randomly permuted and the model has to reconstruct the original order by maximizing the expected log-likelihood over all possible permutations of the input. This task aims to train the model to capture bidirectional context to predict all the tokens instead of just one, which makes it more challenging than MLM. This task is applied in XLNet Z. Yang et al. (2019) and its domain variants ClinicalXLNet Huang et al. (2020).

Causal language modeling (CLM) This is another name for the traditional autoregressive language modeling task, i.e., the model is trained to predict the next token given the previous tokens of the sentence. This task is typically used in autoregressive language models, e.g., GPT Radford et al. (2018) and MedGPT Kraljevic et al. (2021).

Sequence-to-sequence MLM This pre-training task is similar to MLM but performed in a sequence-to-sequence manner. Essentially, the input of the encoder is the corrupted sentence where random tokens are replaced by sentinel tokens, and the target is to make the decoder generate the masked tokens in an autoregressive fashion. This task is adopted in MASS Song, Tan, Qin, Lu, and Liu (2019) and T5 Raffel et al. (2020), and inherited in their domain variants SciFive Phan et al. (2021) and ClinicalT5 Lu, Dou, and Nguyen (2022).

Denoising Autoencoder (DAE) This pre-training task aims to reconstruct the original sentence from a corrupted version of it, where any type of document

corruption functions can be applied such as token masking, token deletion, text infilling, sentence permutation, document rotation, etc. M. Lewis et al. (2020). Essentially, the decoder reconstructs the corrupted input sentence from the output representations of the encoder (i.e., a denoising autoencoder), and the model essentially combines bidirectional and autoregressive transformers. This task is similar to some extent to seq2seq MLM in the sense that they both involve masking out or distorting a portion of the input text and then trying to predict or reconstruct that portion. This task is used in BART M. Lewis et al. (2020), BioBART H. Yuan et al. (2022).

Document Relation Prediction (DRP) This is a novel pre-training task introduced by a recent study Yasunaga, Leskovec, and Liang (2022). Essentially, this task aims to learn the relevance and existence of bridging concepts between documents by classifying the text segment pairs into contiguous, random, or linked. This task can be considered a variation of NSP.

Other Tasks There are other pre-training tasks that are used in specific clinical PLMs. For example, MedGTX S. Park et al. (2022) claims to be the first work to propose graph-text multi-modal pre-training on EHR data. Essentially, they use a Graph Attention Networks (GAT) Velickovic et al. (2017) based encoder to encode the structured information of an EHR, use a BERT-like model to encode the unstructured information (clinical notes), use a cross-model encoder to learn a joint representation space. Moreover, recent studies try to encode domain knowledge into PLMs. For example, UmlsBERT Michalopoulos et al. (2020) continually pre-trains ClinicalBERT Alsentzer, Murphy, Boag, Weng, Jindi, et al.

(2019) on MIMIC-III notes with a specifically designed multi-label loss to inject UMLS knowledge into the model.

2.4 Downstream Tasks

In this section, we introduce the downstream tasks in the clinical domain, along with the corresponding datasets, that have been widely used in recent years. We first discuss the intrinsic tasks, including information extraction, text classification, word/sentence similarity, question answering, text summarization, natural language inference, etc. Then we introduce some popular extrinsic tasks, such as patient readmission prediction, mortality prediction, diagnosis prediction, and other clinical predictive tasks. It is worth noting that the distinction between intrinsic and extrinsic tasks is not always black and white, as some tasks can be considered as both intrinsic and extrinsic, e.g., text-based readmission prediction Lu, Nguyen, and Dou (2021).

2.4.1 Intrinsic Tasks.

Intrinsic tasks are tasks that are primarily focused on understanding the meaning and structure of the text. These tasks are not necessarily the ones that are directly applicable to a specific domain. Examples of intrinsic tasks include, but are not limited to: information extraction, text classification, semantic textual similarity, question answering, text summarization, natural language inference, and others.

Named Entity Recognition Named Entity Recognition (NER) is the most popular downstream NLP task in the clinical domain for the last few years, according to a recent survey Y. Gao et al. (2022). The task refers to identifying and classifying named entities in text into pre-defined categories such as person names, organizations, locations, medical codes, etc, and it is particularly useful for extracting structured information from unstructured text. As a specific application

of NER in the clinical domain, Clinical Named Entity Recognition (CNER) aims to extract clinically relevant information, such as diseases, symptoms, treatments, medications, etc., from unstructured medical texts, e.g., clinical notes in EHRs.

A typical solution to NER is to fine-tune the PLMs to classify each token into one of the pre-defined named entity classes with a linear layer (or more advanced structures such as a LSTM layer) on top of the PLMs. This approach is often referred to as a sequence labeling task. This task has been used for evaluation for a variety of clinical and biomedical PLMs, including BioBERT Lee et al. (2020), SciBERT Beltagy et al. (2019), PubMedBERT Gu et al. (2021), etc.

Relation Extraction As is the case with NER, Relation Extraction (RE) is one of the fundamental IE tasks in the clinical scenario. Essentially, the task refers to identifying and extracting semantic relationships between two or more entities from unstructured text. And in the clinical domain, as a specific application of RE, Clinical Relation Extraction (CRE) aims to extract clinically relevant relationships between medical entities, such as causal relationships (e.g., Patient's high blood pressure caused by obesity.), symptom-disease relationships, medication-disease relationships, etc. depending on the specific task and context.

Essentially, the task is often cast as a classification problem. For example, a common approach to CRE is to fine-tune the PLMs to predict the relationships between two identified entities based on the contextual representations of the `[CLS]` token Su and Vijay-Shanker (2020); Thillaisundaram and Togia (2019).

Event Extraction Event Extraction (EE) is the task that aims to identify and extract event information from text. An event can be defined as a situation or occurrence that happens at a certain point in time and has a specific set of

actors, actions, and outcomes. In text, events are often described using verbs or verb phrases, and the entities involved in the event are typically described using nouns or noun phrases. For example, given a sentence "On Sunday, a protester stabbed an officer with a paper cutter.", a EE system should be able to identify an `Attack` event which consists of an event trigger `stabbed` and event arguments `Sunday, protester, officer, paper cutter` J. Liu, Chen, Liu, Bi, and Liu (2020).

Similarly, Clinical Event Extraction (CEE) is a specific application of EE in the clinical domain, which aims to extract medical events from clinical text, e.g., EHRs. Medical events are occurrences or situations that happen in the medical domain, such as diagnoses, treatments, admissions, etc. For example, a CEE system should extract from the sentence "Patient diagnosed with pneumonia." an event with `diagnosed` as the trigger and `Patient, pneumonia` as the arguments. Event extraction is a challenging task, especially in the clinical domain, due to the complex and private nature of this field. There have been several biomedical event extraction studies in recent years, including DeepEventMine Trieu et al. (2020), BEESL Ramponi, van der Goot, Lombardo, and Plank (2020), etc.

Entity Linking Entity Linking (EL) is a task that aims to link the entity mention in a text to its corresponding entity in a knowledge base, e.g., Wikipedia H. Jiang et al. (2021); Lu and Du (2017); Lu, Gurajada, et al. (2022). In the clinical domain, the task is also referred to as Medical Concept Normalization, which maps medical terms and concepts used in clinical text to a standardized terminology, such as SNOMED CT, ICD-10, or UMLS. There are some tools for this task, e.g., MetaMap Aronson and Lang (2010), SciSpacy Neumann, King, Beltagy, and Ammar (2019), etc.

Coreference Resolution Coreference Resolution is the task of identifying mentions in a text that refer to the same entity. This task is important for a wide range of NLP applications, such as information extraction, machine translation, and question answering, as it helps to understand the structure of the context and to capture the relationships between entities. In the clinical domain, coreference resolution is utilized in analyzing clinical notes, helping to support the decision-making of healthcare professionals by presenting a holistic picture of the patient and the relationships among relevant entities.

Temporal Information Extraction Temporal Information Extraction (TIE) is a task that aims to extract events or facts in the text and link them to specific times. Essentially, this task involves recognition of events and temporal expressions, recognition of temporal relations among them, and timeline construction Leeuwenberg and Moens (2018). TIE in the clinical domain (CTIE) aims to extract temporal information from the clinical text to understand detailed clinical observations.

De-identification : This task is to extract and mask Personal Identifiable Information (PII) from clinical notes, in order to protect patient privacy. The extracted information includes details like patient name, address, Social Security number, etc. This is a particularly important task in the clinical domain as the clinical data must comply with the Health Insurance Portability and Accountability Act (HIPAA).

Text Classification Text Classification is the second most popular downstream task in the clinical domain in recent years Y. Gao et al. (2022).

Essentially, it aims to classify input text into pre-defined categories, such as text-based readmission prediction where they propose to predict ICU patient readmission risk using the clinical notes in EHRs Lu, Nguyen, and Dou (2021).

Semantic Textual Similarity Semantic Textual Similarity (STS) refers to the task of predicting the degree of semantic similarity between words or sentences. The task is useful for a wide range of applications in the clinical domain, as it helps to remove redundant information that could decrease the cognitive load and enhance the clinical decision-making process Y. Wang et al. (2020). Typically, PLMs are used to encode the word/sentence pairs and the cosine distance is used to measure the similarity score.

Question Answering Question Answering (QA) is a task that aims to extract and generate a natural language answer to a given question. Essentially, there are Extractive QA which extracts the answer from the input text, and Open/Closed Generative QA which directly generates a free-text answer to the question based on the input text. Clinical Question Answering (CQA) is a specific application of QA in the clinical domain, and it generates answers to questions related to medical information, such as diagnosis, treatment, medication, etc. CQA systems can be useful in a variety of scenarios, such as hospitals, clinics, and research institutions, to help physicians, nurses, and other healthcare professionals quickly access information and make informed decisions. CQA (or medical QA) is a challenging task as it demands comprehension of medical context, recall of appropriate medical knowledge, and reasoning with expert information Singhal et al. (2022).

There has been a surge of interest in developing PLMs that are capable of answering questions automatically. Recently, Med-PaLM Singhal et al. (2022)

achieves state-of-the-art results on multiple medical QA benchmarks, surpassing previous models including BioMedLM, DRAGON Yasunaga, Bosselut, et al. (2022), BioLinkBERT Yasunaga, Leskovec, and Liang (2022), Galactica Taylor et al. (2022), PubMedBERT Gu et al. (2021), etc. Meanwhile, ChatGPT[4] has attracted huge attention across the world and has demonstrated superior performance over a variety of tasks, leading to a new direction for NLP research.

Text Summarization Text Summarization is the task of extracting the key information of a document and generating a shorter version of it. Similar to other tasks, Clinical Text Summarization refers to the specific application of text summarization in the clinical domain, e.g., clinical notes in EHRs, etc. There are various techniques for text summarization, including extractive summarization and abstractive summarization. Extractive summarization refers to selecting and extracting the most important sentences or phrases from the original text, while abstractive summarization refers to generating a new and shorter text that summarizes the original text.

Natural Language Inference Natural Language Inference (NLI) is a task that aims to predict the relationship between two sentences, i.e., a premise and a hypothesis. The goal of NLI is to classify the relationship between them as either "entailment", "contradiction", or "neutral". Clinical Natural Language Inference (CNLI) is a specific application of NLI in the clinical domain, with the goal of classifying the relationship between two pieces of clinical text. For example, given the premise "Patient has a history of hypertension and diabetes" and the hypothesis "The patient has a high risk of heart disease," the CNLI system should

[4]https://chat.openai.com/chat

predict the relationship as "entailment" as the hypothesis logically follows from the premise. However, if the premise is "Patient has a history of taking aspirin for pain relief" and the hypothesis is "The patient is allergic to penicillin," the relationship should be "neutral" as there is no logical relationship between them. This task is usually cast as a ternary classification problem.

2.4.2 Extrinsic Tasks.

Extrinsic tasks are tasks that are primarily focused on using the understanding of the text to make predictions or decisions in a specific domain. These tasks are more focused on practical or real-world problems or aspects in the specific domain. Examples of extrinsic tasks in the clinical domain include, but are not limited to: readmission prediction, mortality prediction, length of stay prediction, diagnosis prediction, and others Lu et al. (2019); Lu, Dou, and Nguyen (2021b).

2.5 Discussion

2.5.1 Limitations.

Insufficient Domain Expertise There have been tremendous efforts in producing stronger, faster, and larger domain-specific pre-trained language models in the clinical domain. However, most of these models depend on self-supervised pre-training over large amounts of textual data, e.g., ChatGPT uses 175 billion parameters and Med-PaLM has 540 billion parameters Singhal et al. (2022). Recently, ChatGPT has attracted attention all over the world as the model shows remarkable performance on different kinds of NLP-related tasks across multiple domains, including the biomedical and clinical fields. However, the model is still considered "unhelpful" for medicine as judged by human experts as against other domains, revealing that the seemingly almighty model lacks an in-depth understanding of domain knowledge Guo et al. (2023). In fact, there has been

a surge of interest in proposing novel methods to inject domain knowledge into existing PLMs Y. He et al. (2020a); Lu, Dou, and Nguyen (2021a); Michalopoulos et al. (2020). Nevertheless, these works mostly focus on empirical improvement over different benchmarks without providing an in-depth and clear explanation of how the infused knowledge actually affects the model inference, which could limit their impact.

Data Scarcity Another limitation of the clinical PLMs is the limited availability of their pre-training data. Essentially, most of the aforementioned clinical PLMs depend on clinical notes, e.g., the MIMIC database Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019); Huang et al. (2020), which is relatively small in size and does not support the training of larger models Johnson et al. (2016). This scarcity of data can negatively impact the performance of the models and limit their ability to generalize to real-world scenarios.

Interpretability Despite the impressive performance of clinical PLMs, their lack of interpretability remains an issue, as it can limit the trust placed in the models and their ability to be used in real-world clinical settings.

Privacy, Security and Ethical considerations Clinical PLMs often work with sensitive patient information, making privacy and security a major concern. There is a need to ensure that patient data is protected and kept confidential, which can be challenging in the context of Clinical NLP. The use of clinical PLMs also raises important ethical considerations, such as the potential for algorithmic bias and discrimination, the responsibility for the outputs of the models, and the potential impact on patient care and outcomes.

2.5.2 Future Directions.

One promising avenue of future research is to investigate novel pre-training methods that incorporate large amounts of domain knowledge from knowledge bases and limited amounts of clinical notes. The "big knowledge, small data" approach may provide a solution to the challenges of insufficient domain expertise and data scarcity that are faced by current clinical PLMs.

Another important direction is to delve deeper into the interpretability issue of clinical PLMs and their applications. Understanding the thought process and reasoning behind physician diagnoses can provide valuable insights into the use of clinical PLMs. Furthermore, exploring the impact of diverse sources of domain knowledge on model inference can help to better understand how to effectively incorporate knowledge into clinical PLMs. This can lead to improved model performance and increased trust in applying machine learning techniques in the clinical setting.

2.6 Summary

In this chapter, we provide a comprehensive overview of pre-trained language models in the clinical domain. We begin by introducing the key concepts of pre-training methods, model architectures, pre-training data, and other relevant information. Next, we present an extensive list of current clinical PLMs, highlighting their key features and characteristics. Finally, we delve into the limitations of current clinical PLMs, including issues related to the lack of domain knowledge and data scarcity. Finally, we conclude by exploring future directions for Clinical NLP, including the development of novel pre-training methods and a deeper understanding of model interpretability and its applications in the clinical setting.

CHAPTER III

HARNESSING KNOWLEDGE GRAPHS: INTEGRATION TECHNIQUES FOR LANGUAGE MODELS IN HEALTHCARE

This chapter contains materials from the published papers "*Qiuhao Lu, Nisansa de Silva, Sabin Kafle, Jiazhen Cao, Dejing Dou, Thien Huu Nguyen, Prithviraj Sen, Brent Hailpern, Berthold Reinwald, and Yunyao Li. '**Learning electronic health records through hyperbolic embedding of medical ontologies.**' In Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, pp. 338-346. 2019*", "*Qiuhao Lu, Nisansa De Silva, Dejing Dou, Thien Huu Nguyen, Prithviraj Sen, Berthold Reinwald, and Yunyao Li. '**Exploiting node content for multiview graph convolutional network and adversarial regularization.**' In Proceedings of the 28th International Conference on Computational Linguistics, pp. 545-555. 2020*", and "*Qiuhao Lu, Thien Huu Nguyen, and Dejing Dou. '**Predicting patient readmission risk from medical text via knowledge graph enhanced multiview graph convolution.**' In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1990-1994. 2021*". In these publications, the experiments were conducted solely by the author of the dissertation, Qiuhao Lu. The other co-authors provided feedback regarding the results. Qiuhao took complete responsibility for writing all the papers, while Dejing Dou and Thien Huu Nguyen contributed significantly by offering editorial feedback to enhance their quality.

In the previous chapter, we provide an in-depth review of existing clinical pre-trained language models, scrutinizing their architectures, training data, and

more. However, these models, despite exposure to clinical text during pre-training, still struggle with limited domain expertise, hindering their performance in domain-specific tasks. As a solution, this chapter explores the infusion of domain-specific knowledge into these models, with an emphasis on knowledge graphs as a key knowledge source.

In particular, this chapter delves into the utilization of graph representation learning techniques to enhance machine learning models through the integration of both internal and external knowledge graphs in clinical settings. It focuses on three key studies that showcase the potential of knowledge graph integration in clinical applications.

Firstly, we propose a novel approach that incorporates information from EHRs by utilizing hyperbolic embeddings of medical ontologies (with specific reference to ICD-9), within the prediction model Lu et al. (2019). Our results demonstrate the efficacy of this approach, highlighting the promising performance achieved by leveraging hyperbolic embeddings of ontological concepts in clinical applications.

The second study introduces a cutting-edge network embedding method that captures consistency across multiple network views Lu et al. (2020). To achieve this, we generate a secondary view from the input network that reflects node relationships based on content and enforce consistency between the two views by incorporating a multiview adversarial regularization module. Experimental studies conducted on benchmark datasets validate the effectiveness of our method, showcasing superior performance compared to state-of-the-art algorithms in demanding tasks such as link prediction and node clustering. Moreover, when applied to a real-world scenario involving the prediction of 30-day unplanned ICU

readmissions, our method demonstrates promising results in comparison to various baseline approaches.

Lastly, we propose a novel approach that leverages the medical text within EHRs for the prediction of patient outcomes, providing an alternative to prior research that predominantly relied on numerical and time-series patient features Lu, Nguyen, and Dou (2021). Specifically, we extract patients' discharge summaries from EHRs and represent them as multiview graphs, which are further enriched by incorporating an external knowledge graph. Graph convolutional networks are then employed for representation learning. Experimental results validate the effectiveness of this method, showcasing state-of-the-art performance for the given task.

3.1 Learning Electronic Health Records through Hyperbolic Embedding of Medical Ontologies

Patients who are readmitted to intensive care units (ICU) after transfer or discharge are at high risk of mortality, and readmissions are usually costly for both patients and hospitals. Therefore, efficiently and accurately identifying patients who are prematurely discharged or transferred from ICU can not only reduce the risk of mortality but also help decrease the high but avoidable costs of healthcare. According to a recent study Baechle, Agarwal, Behara, and Zhu (2017), unplanned hospital readmission was estimated to have cost nearly $26 billion annually in the U.S. In addition to hospital readmission, ICU readmission is also a major problem. Around 10% of ICU patients are readmitted during the same hospitalization Ponzoni et al. (2017), due to premature discharge or premature transfer from ICU. This highlights the importance of predicting the ICU readmission risk for healthcare systems.

In the past few years, there have been several published studies Krompaß, Esteban, Tresp, Sedlmayr, and Ganslandt (2015); Lin, Zhou, Faghri, Shaw, and Campbell (2019); Rumshisky et al. (2016); Y. Xue, Klabjan, and Yuan (2018) on this unplanned readmission prediction task. Most of the studies are conducted by physicians and medical researchers, and they generally focus on selecting statistically significant features from ICU patients' Electronic Health Records (EHRs) and combining them with traditional machine learning methods, such as logistic regression Y. Xue et al. (2018). These studies prove effectiveness by achieving good prediction accuracy, but they still can be improved by incorporating more sophisticated features, such as the latent embeddings of ontological medical concepts in the patients' EHRs.

Similar to the unplanned readmission prediction task, for in-hospital mortality prediction, there are studies Harutyunyan, Khachatrian, Kale, and Galstyan (2017); Johnson, Kramer, and Clifford (2014) that outperform the traditional scoring systems JR, S, and F (1993); WA, JE, DP, EA, and DE (1981) with machine learning methods. However, they have common limitations: They are not using any external knowledge to improve their models. Therefore, their approaches can be improved by incorporating external knowledge such as medical ontologies.

Medical ontologies are primarily characterized by hierarchical relationships and textual descriptions, along with non-hierarchical features. While Euclidean space is the default geometry for word-based embedding methods Mikolov, Sutskever, Chen, Corrado, and Dean (2013), embeddings learned in hyperbolic spaces Nickel and Kiela (2017) are capable of representing the hierarchies more efficiently. Although there has been some progress in learning word embeddings

in hyperbolic spaces Dhingra, Shallue, Norouzi, Dai, and Dahl (2018), effective leveraging of hierarchies with other data sources remains an open problem.

Hyperbolic space-based representation learning provides an effective way to learn latent embeddings for medical ontologies, which are inherently hierarchical in nature. This significantly helps solve the problem of learning medical ontology embeddings by providing more efficiency and lower dimensions. However, this also poses a significant challenge to medical applications. It is a well-discovered fact that in Euclidean spaces, the learned embeddings are the function of context, which is defined during training. When syntactic structures are taken as the context, words are considered semantically similar when they are surrounded by that same context. Comparable analogies can also be drawn for medical concepts. For example, medical concepts are used by medical providers for billing purposes; thus, similar concepts for such tasks are concepts that co-occur in a diagnosis as well as in a similar hierarchical structure.

In this study, we propose a new method to leverage latent information in the textual data from ICU patients' EHRs, by training and combining the hyperbolic embeddings of the medical concepts in them. We implement our method based on the state-of-the-art method Lin et al. (2019) on ICU readmission prediction and the widely accepted benchmark Harutyunyan et al. (2017) on in-hospital mortality prediction, and we show improvement in both tasks. We also evaluate the hyperbolic embeddings of medical concepts by comparing them with other popular graph embedding methods, both intrinsically and extrinsically. All the experiments are conducted on the MIMIC-III dataset Johnson et al. (2016).

Our contributions are summarized as follows:

– Our method of adding embeddings of ICD-9 codes from discharge summaries proves effective and it helps improve the performance of the state-of-the-art method on ICU readmission prediction with different graph embeddings. It also outperforms the benchmark results in the task of mortality prediction.

– We prove that the hyperbolic embeddings of medical concepts give promising performance in different evaluations, outperforming Euclidean-based graph embeddings in intrinsic evaluation and give comparable performance in extrinsic evaluation.

3.1.1 Related Work.

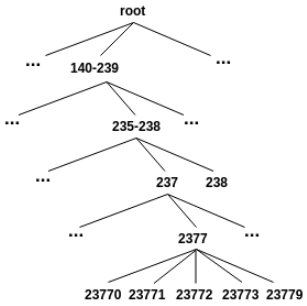
Hyperbolic Representation Learning Representation learning is one of the fundamental characteristics of deep learning advances, with representation learning of words as vectors enabling significant advantages over traditional feature engineering methods. While it is common to use the output of the layer before the last layer of a Convolutional Neural Network (CNN) as an image representation for downstream tasks, it is no surprise to see similar approaches for representation learning being applied to other data sources such as knowledge bases (KBs) Bordes, Usunier, Garcia-Duran, Weston, and Yakhnenko (2013). Recently, a similar idea, in which representation learning of medical concepts which are represented in a large scale KB (e.g., UMLS, SNOMED) Choi, Chiu, and Sontag (2016) are included in the process, has been applied to the medical domain. This has enabled the application of deep learning advances into the medical domain and at the same time increased the range of applications of medical KBs.

It has been shown that linear representations need a much higher number of dimensions in order to represent the hierarchies, which is the most common

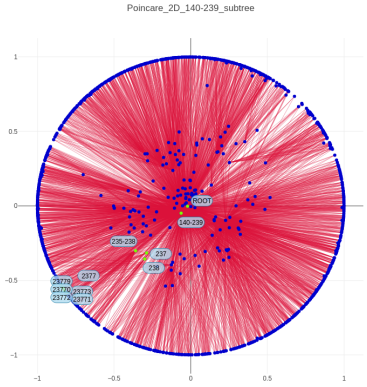
aspect in the KB Nickel and Kiela (2017). Consequently, representation learning in hyperbolic spaces as opposed to Euclidean spaces has been proposed, with hyperbolic space embedding found to perform better in the representation of hierarchies, especially with a lower dimension of features Sala, De Sa, Gu, and Ré (2018). There have been several methods proposed for learning representations in hyperbolic spaces, where it has also been found that data that are not inherently hierarchical (e.g., words) do not yield significant performance gain in hyperbolic spaces Dhingra et al. (2018); Leimeister and Wilson (2018).

Medical ontologies are usually hierarchically organized. This kind of tree-like structure can be well represented in hyperbolic space. To better illustrate, we visualize the Poincaré embedding by training a 2-D embedding of a subtree of the ICD-9 ontology Slee (1978). As shown in Figure 3, the embedding looks like a continuous version of a "tree" and the low-level nodes (leaf) are on the edge and the high-level nodes (root) are on the center area. This is consistent with the feature of hyperbolic space.

Unplanned ICU Readmission Prediction Unplanned ICU readmission prediction, along with unplanned hospital readmission prediction, is an important task in the healthcare field. Apart from research work that is conducted by physicians which is usually based on specific feature engineering and traditional machine learning methods Y. Xue et al. (2018), there is also some solid work that is from the angle of natural language processing which focuses on predicting readmissions based on medical textual notes from EHRs Rumshisky et al. (2016). Some exploit representation learning techniques to solve this problem, where they learn either embeddings of patients or embeddings of medical concepts from patients' data Krompaß et al. (2015); Lin et al. (2019).



(a) Hierarchical Structure of the Subtree



(b) Visualization of the 2-D Hyperbolic Embeddings of the Subtree

Figure 3. Hierarchy and Corresponding 2-D Hyperbolic Embeddings of "140-239" Subtree of ICD-9.

To the best of our knowledge, Lin et al. (2019) proposes the best Area Under the Receiver Operating Characteristics curve (AUROC) score of 0.791 for the ICU readmission prediction task on the MIMIC-III dataset Johnson et al. (2016). They take three types of information as input, i.e., chart events information, basic demographic information and diagnosis information (in the form of ICD-9 codes). All the 3 types of features are concatenated and put into the prediction model, which is a sequential combination of two LSTM layers and one multi-filter CNN layer. They also utilize the embeddings of diagnosis (in the form of ICD-9 codes) to improve the prediction Choi et al. (2016).

Though their work proves effective, they completely overlook the important information that is encoded in the medical text notes in patients' EHRs. Researchers have proved that the medical text notes in patients' EHRs contain enough information that can be used to support the readmission prediction task Rumshisky et al. (2016). In this study, we incorporate the important textual information by extracting ICD-9 codes from the medical notes and embed them into the hyperbolic space, which proves to be a better fit with the ICD-9 medical ontology.

In-hospital Mortality Prediction In-hospital mortality prediction is another important task in the medical domain which aims at predicting the mortality of patients when they are in hospital. Early studies develop systems that calculate the predictions based on expert knowledge or data-driven approaches, such as the APACHE WA et al. (1981), the APACHE II WA, EA, and DP (1985), the SAPS JR et al. (1984), and the SAPS II JR et al. (1993).

Recently, researchers use machine learning techniques to deal with this problem. Johnson *et al.* Johnson et al. (2014) use three traditional methods to solve this task, i.e., Logistic Regression, SVM, and Random Forest. The benchmark Harutyunyan et al. (2017) we use also compares different approaches with traditional scoring systems which includes Logistic Regression and LSTM-based models. Although these works advance the current state-of-the-art performance in mortality prediction, few of them utilize external knowledge like medical ontologies. Hence, in this study, we make use of the ICD-9 codes and represent them with hyperbolic embeddings to see whether it can improve the performance of in-hospital mortality prediction.

ICD-9 and other Medical Ontologies There are multiple medical ontologies:

- **UMLS**: The Unified Medical Language System is a compendium of many controlled vocabularies in the biomedical sciences Bodenreider (2004).
- **SNOMED CT**: SNOMED Clinical Terms is a systematically organized computer processable collection of medical terms providing codes, terms, synonyms and definitions used in clinical documentation and reporting Stearns, Price, Spackman, and Wang (2001).
- **ICD-9**: ICD-9 is the 9-th revision of the International Statistical Classification of Diseases and Related Health Problems (ICD), a medical classification list by the World Health Organization (WHO) Slee (1978). Besides ICD-9, more recent versions (i.e., ICD-10 and ICD-11) are widely used as well.
- **MeSH**: Medical Subject Headings (MeSH) is a comprehensive controlled vocabulary for the purpose of indexing journal articles and books in the life sciences; it serves as a thesaurus that facilitates searching Lipscomb (2000).

In this study, we conduct experiments on the MIMIC-III dataset, which takes ICD-9 as their coding ontology. ICD-9 Clinical Modification (ICD-9-CM) is a modification of ICD-9. This national variant of ICD-9 is provided by the Centers for Medicare and Medicaid Services (CMS) and the National Center for Health Statistics (NCHS), and the use of ICD codes is now mandated for all inpatient medical reporting requirements.

3.1.2 Method.

3.1.2.1 Hyperbolic Medical Concept Embeddings.

We refer the readers to Dhingra et al. (2018); Leimeister and Wilson (2018); Nickel and Kiela (2017) for a more detailed treatment of hyperbolic spaces and their characterization and differences with respect to the Euclidean space geometry. Any metric space is characterized by the distance between two points, with the distance being defined in hyperbolic space, specifically for Poincaré ball model for two points $u, v \in \mathbb{B}^d$ is

$$d_H(u,v) = \text{arcosh}\left(1 + 2\frac{||u-v||^2}{(1-||u||^2)(1-||v||)^2)}\right) \tag{3.1}$$

For a unit Poincaré ball space, $||u|| < 1$. As is evident from Equation 3.1, the distances between two points near the boundary of Poincaré ball tend to ∞. Also, for a hierarchical structure (e.g., a tree) that is embedded into the space, the root node will be placed in the center area of the space while the leaf nodes will be placed near the boundary area.

In order to learn embeddings from hierarchical medical ontologies, we follow the work of Nickel and Kiela (2017) to use Riemannian-SGD to optimize the loss function:

$$L = \sum_{(u,v)\in S} \log \frac{\exp^{-d_H(u,v)}}{\sum_{v'\in N(u)} \exp^{-d_H(u,v')}} \tag{3.2}$$

where $(u,v) \in S$ is a hierarchical (i.e., subclassof) relationship in a Knowledge Base (KB) S and $N(u) = \{v|(u,v) \notin S\} \cup \{u\}$ is a set of negative examples for u. Equation 3.2 can be observed as a soft ranking loss where related objects should be closer than objects for which we do not observe a relationship.

There are different kinds of medical ontologies that hierarchically organize medical concepts including diseases, articles, medicines, etc. Since we conduct experiments on the MIMIC-III dataset, which encodes disease information of patients based on the 9-th revision of the International Statistical Classification

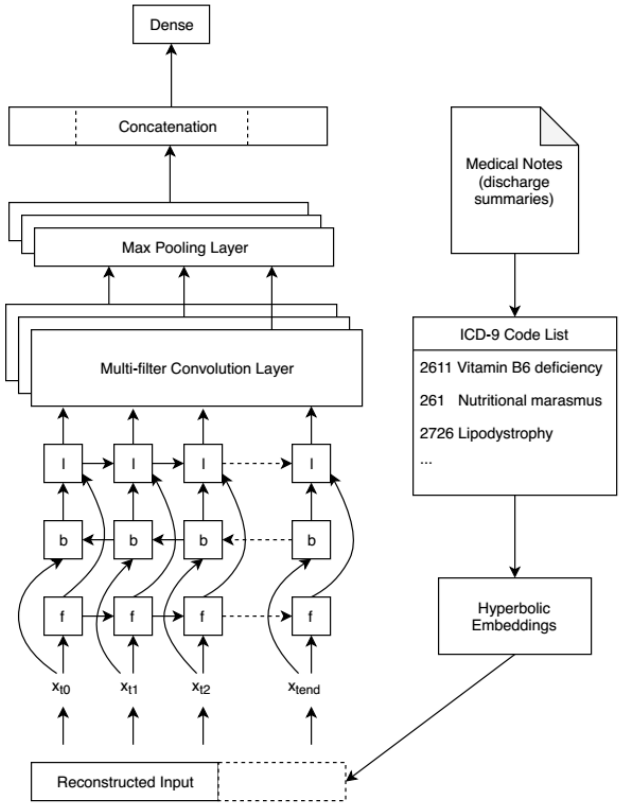


Figure 4. Framework of Readmission Prediction

of Diseases and Related Health Problems (ICD-9), we explore embeddings of the ICD-9 medical concepts for further evaluation.

3.1.2.2 Incorporating Textual Information from EHRs for Readmission Prediction.

In this study, we propose to incorporate the medical text notes, i.e., the discharge summaries, to improve the prediction. We extract ICD-9 codes from the discharge summaries of ICU patients' EHRs using an automatic medical code assignment tool for ICD-9 Perotte et al. (2013). The extracted ICD-9 codes are then embedded into hyperbolic space with the method described in Section 3.1.2.1, the embeddings of which are used to reconstruct the input for the prediction model. Inspired by the use of deep learning models in

Lin *et al.*'s approach Lin et al. (2019), the framework of our method is shown in Figure 4.

Note that in Figure 4, the reconstructed input contains "medical notes ICD-9" which is the embeddings of the list of medical codes extracted from the textual notes (i.e., discharge summaries) in ICU patients' EHRs. Just like the "diagnosis ICD-9" of the original input Lin et al. (2019), they are also in the form of embeddings Choi et al. (2016). But unlike the "diagnosis ICD-9" which is generated manually by professional coders, the "medical notes ICD-9" tends to be redundant but more informative for predicting ICU readmission. Thus, though it is possible that there exists some overlapping between the two lists, our hypothesis is that adding a new list of related ICD-9 codes will help improve the model. The experimental results show that the reconstructed input demonstrates an advantage over the original one Lin et al. (2019).

3.1.2.3 Incorporating Embeddings for Mortality Prediction.

Harutyunyan *et al.*'s work Harutyunyan et al. (2017) is a widely accepted benchmark in in-hospital mortality prediction. We use their benchmark model for our experiment, which is a LSTM network that takes a 48-hour sequence of numerical features (e.g., Glascow coma scale, Heart Rate, etc.) as input. To test our method of incorporating embeddings, we simply concatenate the embeddings of diagnoses of patients (in the form of ICD-9 codes) to the original input and see if any performance gain can be achieved. The framework is shown in Figure 5.

3.1.3 Evaluation.

In this section, we evaluate our proposed method both intrinsically and extrinsically. For intrinsic evaluation, we test different embeddings of the ICD-9 ontology by comparing the similarities between medical concepts in the embedding spaces, to prove that the hyperbolic embedding method

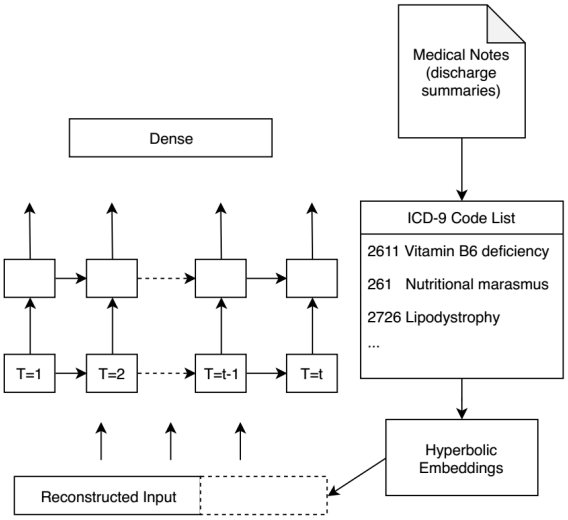


Figure 5. Framework of Mortality Prediction

is a good fit for hierarchical representations, i.e., the ICD-9 ontology. For extrinsic evaluation, we test our method based on the state-of-the-art ICU readmission prediction model Lin et al. (2019) and the in-hospital mortality prediction benchmark Harutyunyan et al. (2017) on the MIMIC-III dataset, with different graph embeddings, to see (1) whether our method improves the performance of ICU readmission prediction and in-hospital mortality prediction; and (2) whether the hyperbolic embeddings of medical concepts from ICD-9 show any advantage over other prevalent embedding algorithms.

3.1.3.1 Intrinsic Evaluation. In this subsection, we intrinsically evaluate the different embeddings over the ICD-9 ontology. Basically, we want to compare and demonstrate how the similarities of medical concepts from ICD-9 are retained in the embedding spaces.

Setup Since we do not have a publicly available gold standard test where the similarities of medical concepts are assigned by professionals, nor the expertise to assign them by ourselves, we take an alternative by randomly selecting a

certain number of pairs of medical concepts from ICD-9 (20,000 in our test), and computing the similarities between them based on several prevalent ontology-based similarity measurements, i.e., the Wu & Palmer similarity Wu and Palmer (1994), the Leacock & Chodorow similarity Leacock and Chodorow (1998), the Resnik similarity Resnik (1995), and the RADA similarity Rada, Mili, Bicknell, and Blettner (1989). Thus, we have 4 sequences of ontology-based term pair similarities over the same set of selected medical concepts.

We then compute the distance-based term pair similarities in the embedding spaces, and we evaluate the embeddings by comparing the Pearson Correlation Coefficients between the sequences of distance-based term pair similarities and the sequences of ontology-based term pair similarities. Note that for the hyperbolic embeddings (Poincaré), we compute the Poincaré distance to denote the *dissimilarity* based on Equation 3.1, and we use the negative value of it to denote the *similarity*. For Euclidean embeddings, we use the Euclidean distance as the *dissimilarity*, and convert it to *similarity* based on $s = \frac{1}{1+d}$.

Intuitively, higher correlation coefficients imply that the similarities between concepts are better retained in the corresponding embedding space.

The 4 ontology-based similarity measurements are defined as follows:

$$\mathrm{Sim}_{\mathrm{WUP}}(C_1, C_2) = \frac{2 * N_3}{N_1 + N_2 + 2 * N_3} \tag{3.3}$$

where N_1 and N_2 are the distance from the least common subsumer (LCS) to C_1 and C_2 respectively. N_3 is the depth of the least common subsumer. The least common subsumer of two concept nodes C_1 and C_2 is the lowest node that can be a parent for C_1 and C_2.

$$\mathrm{Sim}_{\mathrm{LCH}}(C_1, C_2) = -\log\left(\frac{\mathrm{ShortestPath}(C_1, C_2)}{2 * depthmax}\right) \tag{3.4}$$

where *depthmax* is the maximum depth of any node in the tree and ShortestPath(C_1, C_2) is the length of the shortest path between C_1 and C_2.

$$\text{Sim}_{\text{RESNIK}}(C_1, C_2) = \text{IC}(\text{LCS}(C_1, C_2)) \tag{3.5}$$

where LCS refers to the least common subsumer and IC refers to information content. Note that since we cannot compute the term frequency of the medical concepts, we use another ontology-based information content as an alternative Seco, Veale, and Hayes (2004):

$$\text{IC}(c) = 1 - \frac{\log(\text{hypo}(c) + 1)}{\log(maxnodes)} \tag{3.6}$$

where hypo(c) refers to the number of hyponyms of concept c and *maxnodes* refers to the maximum number of concepts in the taxonomy.

$$\text{Sim}_{\text{RADA}}(C_1, C_2) = 2 * depthmax - \text{ShortestPath}(C_1, C_2) \tag{3.7}$$

where *depthmax* and ShortestPath$(C1, C2)$ are the same as Equation 3.4.

Experiments We randomly pick $20,000$ concept pairs from the ICD-9 ontology and compute the mentioned 4 kinds of ontology-based similarities between them. Then we compute the distance-based similarities over these pairs for the several compared embeddings. Finally, we calculate the Pearson Correlation Coefficients between the above two kinds of sequences, as shown in Table 4.

Table 4 shows that the Poincaré embeddings significantly outperform the TransE Bordes et al. (2013), DistMult B. Yang, Yih, He, Gao, and Deng (2014), ComplEx Trouillon et al. (2017); Trouillon, Welbl, Riedel, Gaussier, and Bouchard (2016) and Rescal Nickel, Tresp, and Kriegel (2011) embeddings, in that the Poincaré embeddings show much higher correlation coefficients with the ontology-based similarity sequences. Generally it shows that the similarities

Method	Dim	Measurement			
		WUP	LCH	RESNIK	RADA
Poincaré	10	**0.5720**	**0.6797**	**0.5784**	**0.7278**
	100	**0.5866**	**0.6902**	**0.5977**	**0.7351**
	300	**0.6042**	**0.7046**	**0.6007**	**0.7491**
ComplEx	10	0.4279	0.3169	0.4320	0.3036
	100	0.2265	0.2018	0.2094	0.1774
	300	0.1307	0.1432	0.1134	0.1141
DistMult	10	0.4297	0.3621	0.4174	0.3410
	100	0.1941	0.1827	0.1964	0.1521
	300	0.1204	0.1223	0.1161	0.0922
transE	10	0.0483	0.0269	0.0709	0.0130
	100	0.4159	0.3682	0.3494	0.3658
	300	0.4355	0.3958	0.3862	0.3912
Rescal	10	0.4108	0.2884	0.4364	0.2952
	100	0.2522	0.2756	0.1986	0.2523
	300	0.1243	0.1355	0.1166	0.1039

Table 4. Pearson Correlation Coefficients for Different Embeddings of ICD-9

between concepts are better retained in the hyperbolic embedding space than in the other embedding spaces.

Table 4 also demonstrates that the Poincaré embeddings are capable of representing information with very few dimensions. As shown in this table, the Poincaré embeddings with low dimensions give good performance, similar to the ones with higher dimensions. It thus proves that using hyperbolic-based embedding approaches is a good way to capture semantics in hierarchical data, such as ICD-9.

To sum up, in this subsection we intrinsically evaluate the hyperbolic embeddings over the ICD-9 medical ontology by comparing such with other graph embedding methods. The experimental results demonstrate that the method works well and outperforms other embedding approaches.

Embedding	Acc	Pre-0	Pre-1	Re-0	Re-1	**A.R**	A.P
Poincaré	0.7223	0.9035	0.3740	0.7361	0.6655	0.7786	0.4827
ComplEx	0.6621	0.9141	0.3306	0.6423	0.7454	0.7591	0.4236
Distmult	0.6426	0.9126	0.3172	0.6169	0.7508	0.7534	0.4243
TransE	0.7254	0.9062	0.3789	0.7366	0.6782	**0.7876**	0.4875
Rescal	0.6544	0.9160	0.3264	0.6303	0.7562	0.7661	0.4456

*Acc: Accuracy, Pre: Precision, Re: Recall, A.R: AUC under ROC, A.P: AUC under PRC

Table 5. Performance on ICU Readmission Prediction Without Discharge Summaries

Embedding	Acc	Pre-0	Pre-1	Re-0	Re-1	**A.R**	A.P
Poincaré	0.7481	0.8993	0.4005	0.7766	0.6310	0.7851	0.4819
ComplEx	0.6705	0.9101	0.3342	0.6565	0.7263	0.7602	0.4341
Distmult	0.6678	0.9067	0.3303	0.6565	0.7151	0.7606	0.4327
TransE	0.7536	0.9039	0.4100	0.7779	0.6511	**0.7882**	0.4957
Rescal	0.6399	0.9209	0.3197	0.6067	0.7801	0.7684	0.4454

*Acc: Accuracy, Pre: Precision, Re: Recall, A.R: AUC under ROC, A.P: AUC under PRC

Table 6. Performance on ICU Readmission Prediction With Discharge Summaries

3.1.3.2 Extrinsic Evaluation 1: 30-day Unplanned ICU Readmission Prediction.

Readmission Prediction. In this subsection, we evaluate our proposed method described in Section 3.1.2.2, to see whether any performance improvement on 30-day unplanned ICU readmission prediction can be gained. Moreover, since we apply the hyperbolic embeddings of the ICD-9 medical ontology in the proposed method, this subsection can also be regarded as an extrinsic evaluation test for the medical embeddings.

Setup This portion of our experiments is conducted based on the MIMIC-III Critical Care (Medical Information Mart for Intensive Care III) Database, which is a large, freely-available database composed of deidentified health-related EHR data

Embedding	dim	Acc	Pre-0	Pre-1	Re-0	Re-1	**A.R**	A.P
Poincaré	100	0.7086	0.8769	0.3410	0.7440	0.5590	**0.7193**	0.4098
Poincaré	10	0.6721	0.8886	0.3214	0.6796	0.6403	0.7165	0.3994
TransE	100	0.7115	0.8787	0.3455	0.7461	0.5655	0.7101	0.4067
TransE	10	0.7218	0.8702	0.3475	0.7710	0.5146	0.7099	0.3930

Table 7. Performance on Readmission Prediction with Different Dimensions of Poincaré Embeddings

Embedding	Acc	Pre-0	Pre-1	Re-0	Re-1	**A.R**	A.P
Poincaré	0.8814	0.8977	0.6350	0.9737	0.2912	0.8722	0.5543
ComplEx	0.8947	0.9165	0.6701	0.9662	0.4380	0.8915	0.6104
Distmult	0.8988	0.9152	0.7115	0.9730	0.4243	**0.8956**	0.6247
TransE	0.8888	0.9019	0.6930	0.9777	0.3211	0.8854	0.5717
Rescal	0.8913	0.9019	0.7239	0.9809	0.3188	0.8954	0.6025

*Acc: Accuracy, Pre: Precision, Re: Recall, A.R: AUC under ROC, A.P: AUC under PRC

Table 8. Performance on Mortality Prediction Without Discharge Summaries

associated with over $40,000$ patients who stayed in the critical care units (ICU) of the Beth Israel Deaconess Medical Center between 2001 and 2012.

The database contains a large variety of EHR data of ICU patients, including basic demographic information, bedside vital sign measurements, laboratory test results, medications, procedures, medical text notes (e.g., discharge summaries), and so on.

In this experiment, we follow the data preprocessing procedure of Harutyunyan et al. (2017); Lin et al. (2019) and generate a dataset of $48,411$ ICU stay records. Each ICU stay record corresponds to one ICU patient, and each patient may have multiple ICU stay records. We then split the entire dataset into the training set (80%), the validation set (10%), and the testing set (10%) for further evaluation.

For fair comparison, we use the same setup and benchmark with Lin*et al.* Lin et al. (2019) and consider 4 types of positive ICU stay records, including the patients (and the corresponding ICU stay record) who were transferred to low-level wards from ICU and readmitted to ICU later; the patients who were transferred out of ICU and died later; the patients who were discharged and readmitted to ICU later; and the patients who were discharged and died later. Note that the "later" here means "within 30 days."

Experiments We experiment with the hyperbolic embeddings (Poincaré) of the ICD-9 ontology and several state-of-the-art graph embedding methods, i.e., ComplEx, Distmult, TransE and Rescal. The results of using our method, with different embeddings, on the ICU readmission prediction task are shown in Table 5 and Table 6.

Note that in the readmission prediction task, most researchers are using the Area Under the Receiver Operating Characteristics curve (AUROC) as the main metric to evaluate their approaches. Generally, a higher AUROC score means a better model, for this task. Along with AUROC (A.R), some additional metrics are proposed, to better illustrate the comparison. However, these additional metrics can be unstable, and they are better used for additional evaluation.

In Table 5, we present the performance of different embeddings without ICD-9 codes extracted from discharge summaries. Note that in Table 5 we only use the human-annotated ICD-9 codes for each patient, without using any extractions from the discharge summaries. In Table 6, we present the corresponding results with ICD-9 codes extracted from discharge summaries as described in Section 3.1.2.2. It shows that adding extra ICD-9 codes from discharge summaries does improve the overall performance on this readmission prediction task. It

also shows that the Poincaré embeddings outperform all other graph embedding methods except TransE. Note that we also test this method with the Claims embeddings Choi et al. (2016) that are used by Lin *et al.* Lin et al. (2019), the results of which (0.7943) also demonstrate an advantage over their best reported A.R score (0.791). We do not think it is fair to compare Lin *et al.*'s results Lin et al. (2019) with the reported graph embedding methods in Table 5 and 6 because they only use ICD-9 codes to generate embeddings.

As is described in Section 3.1.1, hyperbolic embeddings have the ability to represent hierarchical data with lower dimensions. So, in Table 7, we test our method using the Poincaré and TransE embeddings with lower dimensions. The results are consistent, showing that lower dimensions of Poincaré embeddings give better performance than that of TransE, especially when in 300 dimensions TransE actually does better than Poincaré.

To sum up, in this subsection we evaluate our method in the ICU readmission prediction task, and we also extrinsically evaluate the hyperbolic embeddings of the ICD-9 ontology. The results prove the effectiveness of our method by showing a better AUROC over the model without discharge summaries. The results also demonstrate the good qualities of the hyperbolic embeddings, in that they give comparable performance with the state-of-the-art graph embedding methods.

3.1.3.3 Extrinsic Evaluation 2: In-Hospital Mortality Prediction.

In this subsection, we further evaluate our method and the hyperbolic embeddings of the ICD-9 medical ontology by incorporating the embeddings into existing methods of in-hospital mortality prediction and comparing their performance.

Embedding	Acc	Pre-0	Pre-1	Re-0	Re-1	**A.R**	A.P
Poincaré	0.8882	0.9128	0.6338	0.9626	0.4128	0.8756	0.5760
ComplEx	0.8972	0.9012	0.8220	0.9895	0.3073	0.8958	0.6312
Distmult	0.8941	0.9267	0.6330	0.9529	0.5183	0.8959	0.6218
TransE	0.8882	0.8995	0.7043	0.9802	0.3004	0.8852	0.5771
Rescal	0.8929	0.9141	0.6654	0.9669	0.4197	**0.8979**	0.6042

*Acc: Accuracy, Pre: Precision, Re: Recall, A.R: AUC under ROC, A.P: AUC under PRC

Table 9. Performance on Mortality Prediction With Discharge Summaries

Embedding	dim	Acc	Pre-0	Pre-1	Re-0	Re-1	**A.R**	A.P
Poincaré	300	0.8882	0.9128	0.6338	0.9626	0.4128	0.8756	0.5760
Poincaré	100	0.8938	0.9081	0.7061	0.9759	0.3692	**0.8789**	0.5841
Poincaré	10	0.8904	0.9100	0.6627	0.9691	0.3876	0.8755	0.5900

Table 10. Performance on Mortality Prediction with Different Dimensions of Poincaré Embeddings

Setup This part of our experiments is also conducted on the MIMIC-III dataset Johnson et al. (2016). We follow the data preprocessing pipeline with the benchmark Harutyunyan et al. (2017). The data contains $42,276$ ICU stays of $33,798$ unique, de-identified patients, who are at least 18 years old. For fair comparison, we adopt the same split of 15% for validation and 85% for training.

Experiments To be consistent with the experiment on 30-day unplanned ICU readmission prediction, we experiment with the same group of graph embeddings (i.e., Poincaré, ComplEx, Distmult, TransE and Rescal). The results of our method with different embeddings on in-hospital mortality prediction are shown in Table 8 and Table 9.

In the task of in-hospital mortality prediction, Area Under the Receiver Operating Characteristics curve (AUROC) is still the widely accepted metric for

evaluation. Higher AUROC indicates better performance of a model for a certain embedding method. In Table 8 and 9, the same set of metrics is represented for additional comparison.

The work of Harutyunyan *et al.* Harutyunyan et al. (2017) is a widely accepted benchmark for mortality prediction, which gives an A.R score of 0.8607. As in the readmission experiment, we present the results with and without ICD-9 codes extracted from discharge summaries. Note that in Table 8 we only use the human-annotated ICD-9 codes for each patient, without using any extractions from the discharge summaries. It shows that adding ICD-9 codes can generally improve the performance of mortality prediction with different embeddings. Every embedding method in the experiment leads to an improvement on the AUROC (A.R) score over the baseline (0.8607). In Table 10, we test the Poincaré embeddings with different dimensions, and the results are very stable, which is consistent with the earlier assumption.

In summary, we extrinsically evaluate our method and the hyperbolic embeddings of the ICD-9 medical ontology on the task of in-hospital mortality prediction. The results prove the effectiveness of our method by representing higher AUROC than the benchmark, though in this task the hyperbolic embeddings do not outperform all other embeddings. However, adding ICD-9 codes extracted from discharge summaries does improve the overall performance with almost every embedding method on the task of mortality prediction, which is consistent with the readmission prediction experiment.

3.2 Exploiting Node Content for Multiview Graph Convolutional Network and Adversarial Regularization

Over the last few years, network representation learning, or node embedding, has gained increasing interest in the community of machine learning, due to the popularity of the special data form. In reality, datasets from different fields are often in the form of networks, such as social networks, drug-target-interaction networks, mobile phone networks, citation networks, etc. It is therefore very important to find a way to well represent the networks, which is challenging because there is no direct way to encode the high-dimensional data into low-dimensional feature vectors efficiently W. L. Hamilton, Ying, and Leskovec (2017). Moreover, network embedding techniques benefit a variety of downstream applications like link prediction, node classification, and node clustering.

In recent years, researchers have developed different kinds of network embedding approaches, many of which have shown great performance in analytical evaluation and have been quite effective in downstream applications. These studies range from traditional machine learning techniques like matrix factorization to recent deep-learning-based methods like graph autoencoders.

Traditional models, or shallow models, usually optimize the embeddings of nodes directly. For these shallow models, the mapping from networks to vectors is simply an embedding lookup, i.e., each node corresponds to a unique embedding vector W. L. Hamilton et al. (2017). Factorization-based approaches like GraRep Cao, Lu, and Xu (2015), HOPE Ou, Cui, Pei, Zhang, and Zhu (2016) and random walk-based approaches like DeepWalk Perozzi, Al-Rfou, and Skiena (2014), node2vec Grover and Leskovec (2016) all fall into this category. Shallow models

generally suffer from computational inefficiency and lack of ability to well represent complex networks.

More recently, deep models, or autoencoder-based approaches, have been gaining more and more attention, and have shown superior performance in many applications. Compared with shallow models which use a simple lookup table as the encoder function, deep models usually use deep neural networks as the encoder. For example, SDNE D. Wang, Cui, and Zhu (2016) and DNGR Cao, Lu, and Xu (2016) use deep neural networks as the encoder and decoder functions to generate low-dimensional representations. GAE and VGAE Kipf and Welling (2016b) aggregate neighborhood messages based on convolutional encoders, e.g., graph convolutional networks (GCN) Kipf and Welling (2016a) and its variants, to generate node embeddings. The encoders share parameters across nodes and it leads to better efficiency. Note that GCN variants like GraphSAGE W. Hamilton, Ying, and Leskovec (2017) and GAT Veličković et al. (2017) are not discussed as they mostly focus on message passing which is not the main focus of this work.

Another successful variant of graph autoencoders incorporates generative adversarial networks (GAN) for representation learning. For example, ARGA and ARVGA Pan et al. (2018) enforce the latent node embeddings to match a prior normal distribution based on an adversarial training mechanism. The adversarial training procedure usually provides regularization and results in more robust and meaningful representations Makhzani, Shlens, Jaitly, Goodfellow, and Frey (2015). DBGAN Zheng et al. (2020) estimates the prior distribution of latent representations by prototype learning and aims to balance both sample-level and distribution-level consistency via a novel bidirectional adversarial learning framework.

A common theme among most of the aforementioned approaches is that they do not explicitly consider the semantic relatedness between nodes. For shallow models like DeepWalk Perozzi et al. (2014), they mostly only focus on preserving the topological structure of the network while neglecting the rich information in node content. For deep models, they implicitly incorporate node content by aggregating neighborhood node features using powerful encoders like graph convolutional networks.

In this paper, we propose a novel network embedding method based on multiview graph convolutional networks and adversarial regularization. The method aims to preserve the distribution consistency across two views of the network, as well as shape the output representations to match an arbitrary prior distribution, by incorporating a multiview adversarial regularization module. More specifically, we regard the topological structure as the first and main view of the network, and create a second view that captures the relatedness between nodes based on node content. Different from DBGAN Zheng et al. (2020) which tries to reconstruct the node features directly, the proposed method relaxes this requirement and focuses on preserving the semantic relatedness between them. A multiview reconstruction loss function is leveraged to optimize the model jointly. We evaluate the proposed method on three diverse applications. The experimental results on benchmark datasets demonstrate that the method outperforms the state-of-the-art algorithms in link prediction and node clustering. We also evaluate our method on a real-world downstream application, i.e., ICU readmission prediction, and the method compares favorably with several baseline methods. Our contributions can be summarized as follows:

– We propose a novel network embedding method, i.e., Multiview Adversarially Regularized Graph Autoencoder (MRGAE). Unlike previous studies that either neglect node content or aim to reconstruct the entire node feature matrix, we focus on the semantic relatedness between nodes and aim to preserve the consistency of node presentations across two specific views of the network. We incorporate a multiview adversarial regularization module to achieve the objective and enforce the output representations to match a prior distribution.

– We conduct extensive and diverse experiments for evaluation. The experimental studies demonstrate the superb performance of our method, by updating the state-of-the-art results in link prediction and node clustering on benchmark datasets. Our method also compares favorably with baselines in the task of ICU readmission prediction.

3.2.1 Method.

Graph Convolutional Networks Most recent graph neural network models usually use a common architecture, i.e., graph convolutional networks (GCN) Kipf and Welling (2016a), to encode the input networks. Essentially, graph convolutional networks transform the original graph or network into a lower-dimensional representation matrix $\mathbf{Z}$, given the adjacency matrix $\mathbf{A}$ and the feature matrix $\mathbf{X}$ as the input. Each of the transformations can be written as a non-linear convolution function:

$$\mathbf{H}^{(l+1)} = f(\mathbf{H}^{(l)}, \mathbf{A}) \tag{3.8}$$

where $\mathbf{H}^{(0)} = \mathbf{X}$ which is the input feature matrix, and $\mathbf{H}^{(l)}$ refers to the output representation matrix (i.e., embeddings) $\mathbf{Z}^{(l)}$ for the l-th layer convolutional

neural network. Essentially, different types of the convolution function f usually correspond to variants of the GCN model. The standard convolution function can be written as:

$$\mathbf{Z}^{(l+1)} = \mathbf{H}^{(l+1)} = f(\mathbf{H}^{(l)}, \mathbf{A}) = \sigma(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}) \tag{3.9}$$

where $\hat{\mathbf{A}} = \mathbf{A}+\mathbf{I}$, and $\mathbf{I}$ is the identity matrix of $\mathbf{A}$. $\hat{\mathbf{D}}$ is the diagonal degree matrix of $\hat{\mathbf{A}}$, and $\mathbf{W}^{(l)}$ is the weight matrix for the l-th layer neural network, which is also the parameter to optimize. We use the ReLU function as the activation function σ in this paper, and adopt a two-layer GCN as the encoder for all the experiments.

Adversarial Regularization Adversarial regularization has proven effective in various network representation learning approaches Q. Dai, Li, Tang, and Wang (2018); Makhzani et al. (2015); Pan et al. (2018). Generally, in the *encoder-decoder* framework, one can view the encoder as a generator, and incorporate a discriminator (e.g., a multi-layer perceptron) to distinguish whether a latent representation is from the encoder or from an arbitrary prior distribution. By incorporating this module, one can shape the learned representations to match an arbitrary prior distribution, e.g., Gaussian distribution. This is similar in spirit to VGAE, which uses KL divergence instead of adversarial training to achieve the same purpose Makhzani et al. (2015). In this work, we extend the adversarial regularization module to a multiview scenario, where we aim to enforce the learned representations from the two views to be distribution consistent and to match a prior distribution.

Multiview Adversarially Regularized Graph Autoencoder (MRGAE) The overall framework of the proposed method contains three main parts, as

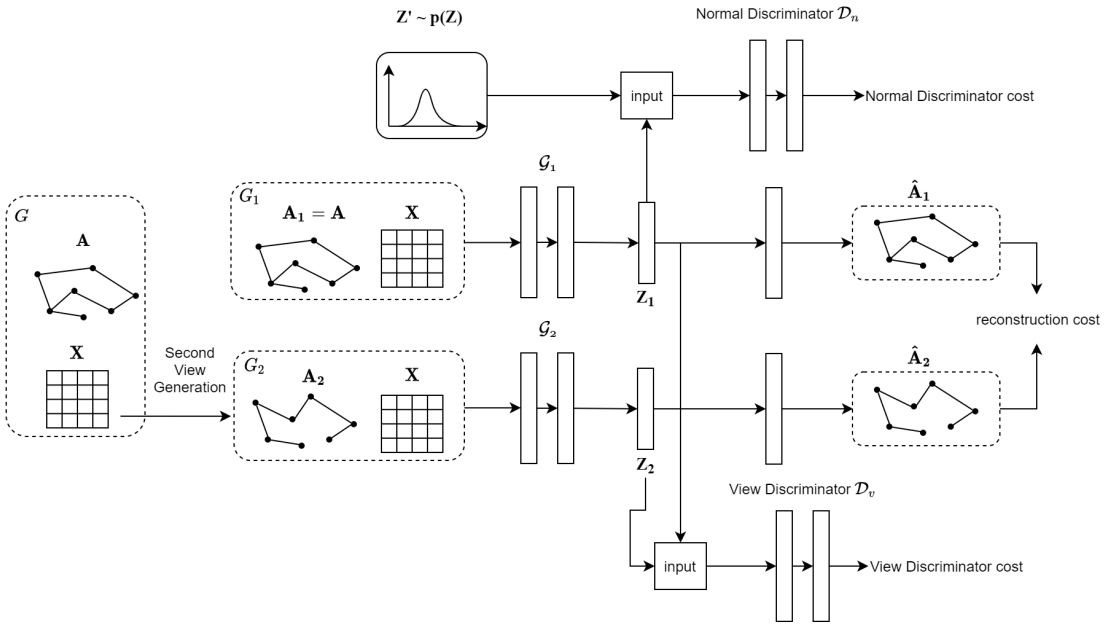


Figure 6. Architecture of MRGAE.

depicted in Figure 6. First, we consider the topological structure as the first and main view of the input network, and create a second view of it. Next, we use two graph convolutional networks (GCN) as the encoders to separately encode the two views of the input network. Then, we incorporate two discriminators, one to distinguish between the representations from the main view and the prior distribution, and the other to distinguish between the representations from the two views. In this paper, we use the Gaussian distribution as the prior distribution since the Gaussian assumption has been widely adopted in various previous studies Kipf and Welling (2016b); Makhzani et al. (2015); Pan et al. (2018). Finally, we design a specific multiview reconstruction loss function, combine it with the two discriminators, and optimize the model jointly.

Notations Given the undirected input network $G = (V, E)$, we regard it as the first view G_1 and create a second view G_2 from it. Specifically, we denote the two

views of the network G as $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, respectively. Note that we have $E_1 = E$. Each view $G_i (i = 1, 2)$ has the same node set V with N nodes ($N = |V|$) and a different set of edges E_i. Each view has its own adjacency matrix $\mathbf{A_i}$ and degree matrix $\mathbf{D_i}$. We further introduce a $N \times D$ feature matrix $\mathbf{X}$ for V, where each row corresponds to the input features of D dimensions for each node. For featureless networks, we use the identity matrix as a replacement for $\mathbf{X}$. The goal is to learn a unified representation matrix $\mathbf{Z}$ for the nodes.

Second View Construction We aim to construct a second view $G_2 = (V, E_2)$ of the network that captures the semantic relatedness between nodes. To define E_2, we adopt a straightforward strategy to calculate cosine similarities between node content. Essentially, if the cosine similarity between two nodes is greater than a threshold α_{prox}, then we create a link between them in the second view.

Encoder-Decoder Framework In this paper, we follow the generalized *encoder-decoder* framework W. L. Hamilton et al. (2017) for learning network representations. More specifically, we adopt two-layer GCNs as the encoders, and each of them encodes one single view of the input multiview network. Essentially, the encoder model transforms the nodes in the network into low-dimensional feature representations (i.e., embeddings), and this encoding process can be written as:

$$\mathbf{Z_i} = \mathrm{ENC}(\mathbf{X}, \mathbf{A_i}) = \mathrm{GCN}(\mathbf{X}, \mathbf{A_i}) \tag{3.10}$$

where $\mathbf{Z_i}$ refers to the representation matrix learned from the i-th view G_i. Along with Equation 3.9, the encoding process can then be further explained as:

$$\mathbf{Z_i^{(0)}} = \mathbf{X} \tag{3.11}$$

$$\mathbf{Z_i^{(1)}} = \text{LeakyReLU}(\hat{\mathbf{D}}_\mathbf{i}^{-\frac{1}{2}}\hat{\mathbf{A}}_\mathbf{i}\hat{\mathbf{D}}_\mathbf{i}^{\frac{1}{2}}\mathbf{X}\mathbf{W_i}^{(0)}) \tag{3.12}$$

$$\mathbf{Z_i^{(2)}} = \hat{\mathbf{D}}_\mathbf{i}^{-\frac{1}{2}}\hat{\mathbf{A}}_\mathbf{i}\hat{\mathbf{D}}_\mathbf{i}^{\frac{1}{2}}\mathbf{Z_i^{(1)}}\mathbf{W_i}^{(1)} \tag{3.13}$$

where $\hat{\mathbf{A}}_\mathbf{i}$ and $\hat{\mathbf{D}}_\mathbf{i}$ refer to the adjacency matrix and degree matrix of the i-th view G_i, respectively. Similarly, $\mathbf{W_i}^{(l)}$ represents the parameter matrix for the l-th layer graph convolutional network with G_i. Thus, in general this encoding process with Equation 3.10 can be written as:

$$\mathbf{Z_i} = \text{ENC}(\mathbf{X}, \mathbf{A_i}) = q(\mathbf{Z_i}|\mathbf{X}, \hat{\mathbf{A}}_\mathbf{i}) = \mathbf{Z_i^{(2)}} \tag{3.14}$$

With regard to the *decoder* model, essentially it decodes the learned low-dimensional representations, and transforms them into some information that can be evaluated in some way, for example, the existence of edges between nodes or label predictions on specific downstream tasks. The evaluations are a good way to measure the quality of the learned representations of nodes. In this paper, we use a simple yet effective pair-wise inner-product decoder to reconstruct the edges of the original network, which is shown as follows:

$$\text{DEC}(\mathbf{z_p}, \mathbf{z_q}) = \mathbf{z_p}^\top \mathbf{z_q} \tag{3.15}$$

The inner-product decoder model aims to reconstruct the edge set between nodes in the input network, where the reconstructed edge set should be as similar as the original one. In our case, the reconstruction loss is calculated based on each of the views, i.e., the decoder aims to reconstruct each view from the learned representations from that view, respectively. The decoding process is shown as follows:

$$p(\hat{\mathbf{A}}_\mathbf{i}|\mathbf{Z_i}) = \prod_{p=1}^{N}\prod_{q=1}^{N} p((\hat{A}_i)_{pq}|\mathbf{z_{ip}}, \mathbf{z_{iq}}) \tag{3.16}$$

$$
\begin{aligned}
p((\hat{A}_i)_{pq} = 1|\mathbf{z_{ip}}, \mathbf{z_{iq}}) &= \sigma_s(\mathrm{DEC}(\mathbf{z_{ip}}, \mathbf{z_{iq}})) \\
&= \sigma_s(\mathbf{z_{ip}}^\top \mathbf{z_{iq}})
\end{aligned}
\tag{3.17}
$$

where $(\hat{A}_i)_{pq}$ refers to the edges between nodes, and σ_s here is the logistic sigmoid function.

Multiview Adversarial Regularization The intuition is that we want the latent embeddings learned from different views are consistent, i.e., the same nodes from different views are close in the embedding space, and the learned latent embeddings from different views fit a similar distribution. Thus, we propose the loss function should be in the following form:

$$
\mathcal{L} = \sum_{i=1}^{2} (\alpha_i \mathbb{E}_{q(\mathbf{Z_i}|\mathbf{X}, \hat{\mathbf{A}}_\mathbf{i})}[-\log p(\hat{\mathbf{A}}_\mathbf{i}|\mathbf{Z_i})]) + \mathcal{S} \tag{3.18}
$$

where α_i are the balancing coefficients. Intuitively, the first term corresponds to the addition of the individual reconstruction loss from each view. The second term $\mathcal{S}$ is the term that models the consistency across different views, and the specific methods differ in how this term is chosen and parameterized.

We then introduce a multiview reconstruction loss (MRL) function:

$$
\mathcal{L}_{mrl} = \sum_{i=1}^{2} (\alpha_i \mathbb{E}_{q(\mathbf{Z_i}|\mathbf{X}, \hat{\mathbf{A}}_\mathbf{i})}[-\log p(\hat{\mathbf{A}}_\mathbf{i}|\mathbf{Z_i})]) + \beta \mathbb{E}_{\mathbf{Z_1} \sim q(\mathbf{X}, \hat{\mathbf{A}}_\mathbf{1}), \mathbf{Z_2} \sim q(\mathbf{X}, \hat{\mathbf{A}}_\mathbf{2})}[-\log p(\hat{\mathbf{A}}_\mathbf{1}|\mathbf{Z_1}, \mathbf{Z_2})] \tag{3.19}
$$

where the first term refers to the addition of the individual reconstruction loss from each view, and the second term is the loss of reconstructing the graph structure of the main view G_1 with the encoded representations from both views, i.e., $\mathbf{Z_1}$ and $\mathbf{Z_2}$. Here instead of only adding the individual reconstruction loss together, we use the encoded representations from both views to jointly reconstruct the main structure, thus achieving better consistency and robustness. More specifically, we have $p((\hat{A}_1)_{pq} = 1|\mathbf{z_{1p}}, \mathbf{z_{2q}}) = \sigma_s(\mathbf{z_{1p}}^\top \mathbf{z_{2q}})$.

Method	Cora		Citeseer		Pubmed	
	AUC	AP	AUC	AP	AUC	AP
SC	84.6 ± 0.01	88.5 ± 0.00	80.5 ± 0.01	85.0 ± 0.01	84.2 ± 0.02	87.8 ± 0.01
DW	83.1 ± 0.01	85.0 ± 0.00	80.5 ± 0.02	83.6 ± 0.01	84.2 ± 0.00	84.1 ± 0.00
GAE	91.0 ± 0.02	92.0 ± 0.03	89.5 ± 0.04	89.9 ± 0.05	96.4 ± 0.00	96.5 ± 0.00
VGAE	91.4 ± 0.01	92.6 ± 0.01	90.8 ± 0.02	92.0 ± 0.02	94.4 ± 0.02	94.7 ± 0.02
ARGA	92.4 ± 0.003	93.2 ± 0.003	91.9 ± 0.003	93.0 ± 0.003	96.8 ± 0.001	97.1 ± 0.001
ARVGA	92.4 ± 0.004	92.6 ± 0.004	92.4 ± 0.003	93.0 ± 0.003	96.5 ± 0.001	96.8 ± 0.001
MRGAE	94.0 ± 0.7	94.1 ± 0.6	94.3 ± 0.4	94.9 ± 0.8	97.2 ± 0.2	97.4 ± 0.3
DBGAN	94.5 ± 0.01	95.1 ± 0.05	94.5 ± 0.04	95.8 ± 0.01	96.8 ± 0.01	97.3 ± 0.02
MRGAE*	$\mathbf{95.0 \pm 0.3}$	$\mathbf{95.2 \pm 0.4}$	$\mathbf{95.7 \pm 0.5}$	$\mathbf{96.4 \pm 0.4}$	$\mathbf{97.8 \pm 0.1}$	$\mathbf{97.8 \pm 0.2}$

Table 11. Performance comparison on link prediction.

Unlike previous work, we incorporate two discriminators, namely the normal discriminator $\mathcal{D}_n$ and the view discriminator $\mathcal{D}_v$, to distinguish between the representations from the main view and the Gaussian distribution, and to distinguish between the representations from the two views, as depicted in Figure 6. We share weights between them. The adversarial loss for the two discriminators is defined as:

$$\begin{aligned}\mathcal{L}_{adv} = &- (\mathbb{E}_{\mathbf{Z_n}\sim\mathcal{N}}[\log \mathcal{D}_n(\mathbf{Z_n})] + \mathbb{E}_{\mathbf{x}\sim p(x)}[1 - \mathcal{D}_n(\mathcal{G}_1(\mathbf{X}, \mathbf{A_1}))]) \\ &- (\mathbb{E}_{\mathbf{x}\sim p(x)}[\log \mathcal{D}_v(\mathcal{G}_1(\mathbf{X}, \mathbf{A_1}))] + \mathbb{E}_{\mathbf{x}\sim p(x)}[1 - \mathcal{D}_v(\mathcal{G}_2(\mathbf{X}, \mathbf{A_2}))])\end{aligned} \tag{3.20}$$

where $\mathcal{G}_1$ and $\mathcal{G}_2$ refer to the two GCN encoders, respectively. And finally, we use a weighted sum of the above losses:

$$\mathcal{L}_1 = \mathcal{L}_{mrl} + \gamma \mathcal{L}_{adv} + \mathcal{L}_{reg} \tag{3.21}$$

where $\mathcal{L}_{reg}$ is a regularization term and we have $\mathcal{L}_{reg} = \mathbb{E}_{\mathbf{Z_1}\sim q(\mathbf{X}, \hat{\mathbf{A}}_\mathbf{1})}[-\log \mathcal{D}_n(\mathbf{Z_1})]$. We then jointly train the model by minimizing $\mathcal{L}_1$, and finally take the encoded representations from the main view, i.e., $\mathbf{Z_1}$, as the output representations.

3.2.2 Experiments. In this section, we evaluate our proposed method based on three tasks. First, we conduct the experiment of link prediction on the benchmark dataset of three citation networks. We also report the experiment of node clustering on these networks. Finally, we apply the proposed method to a real-world medical application, i.e., 30-day unplanned ICU readmission prediction.

3.2.2.1 Link Prediction. Link prediction is a popular task in evaluating network embedding methods. Essentially, a small portion of the edges are removed for generating the validation and test sets, and the same number of pairs of unconnected nodes are randomly picked as negative samples. The goal of the task is to predict whether or not there exists an edge between two nodes.

Dataset and Second View Construction We conduct the experiment on three popular citation networks, i.e., Cora, Citeseer and Pubmed Sen et al. (2008). The nodes represent scientific publications from different areas, and the edges represent the citation links between them. The nodes are represented with feature vectors, which are described by 0/1-valued word vectors indicating the absence/presence of the corresponding word (Cora and Citeseer) or tf-idf weighted word vectors (Pubmed). Each node has a corresponding class label.

In this experiment, we take the original edge set of the input network, i.e., the citation links, as the first and main view. We construct the second view based on textual similarities. Essentially, if the cosine similarity between two publications is greater than the empirical threshold 0.7, then we create a link between them in the second view.

Baselines We compare the proposed method with several baseline methods: Spectral Clustering (SC) Tang and Liu (2011), Deepwalk (DW), Graph

Autoencoder (GAE), Variational Graph Autoencoder (VGAE), Adversarially Regularized Graph Autoencoder (ARGA), Adversarially Regularized Variational Graph Autoencoder (ARVGA) and DBGAN Zheng et al. (2020).

Experiment Settings For all the experiments, we split each of the datasets into the training set (85%), the validation set (5%), and the test set (10%). To reduce the influence of randomness, we average the results over five randomly selected splits as in Zheng et al. (2020).

We use the same set of hyperparameters for the GCN encoder with the baselines Kipf and Welling (2016b); Pan et al. (2018); Zheng et al. (2020). More specifically, we use a 32-dim hidden layer and 16-dim latent representations for the GCN encoder in the link prediction task. We also use two multi-layer perceptrons (MLP) as the discriminators, each of which consists of two 128-dim hidden layers. We set the balancing factors $\alpha_1, \alpha_2, \gamma$ to 1.0, and set β to 0.8 in all experiments. The performance of our method is recorded as MRGAE in Table 11.

Note that DBGAN uses a larger embedding size in their experiments. For a fair comparison, we also set the representation size to 32-dim (Cora) and 64-dim (Citeseer and Pubmed), the results of which are recorded as MRGAE*.

Results We use the same evaluation metrics with the previous work, i.e., *area under the Receiver Operating Characteristics curve* (AUC) and *average precision* (AP) scores.

As shown in Table 11, the proposed method (MRGAE) achieves the best performance on all three citation networks, outperforming the state-of-the-art method, i.e., DBGAN, indicating the effectiveness of exploiting node content by incorporating multiview adversarial regularization.

Method	Acc	NMI	F1	Prec	ARI
SC	0.367	0.127	0.318	0.193	0.031
DW	0.484	0.327	0.392	0.361	0.243
RTM	0.440	0.230	0.307	0.332	0.169
RMSC	0.407	0.255	0.331	0.227	0.090
TADW	0.560	0.441	0.481	0.396	0.332
GAE	0.596	0.429	0.595	0.596	0.347
VGAE	0.609	0.436	0.609	0.609	0.346
ARGA	0.640	0.449	0.619	0.646	0.352
ARVGA	0.638	0.450	0.627	0.624	0.374
MRGAE	0.703	0.523	0.681	0.716	0.476
GALA	0.745	**0.576**	–	–	0.531
DBGAN	0.748	0.560	–	–	0.540
MRGAE*	**0.764**	0.559	**0.740**	**0.742**	**0.570**

Method	Acc	NMI	F1	Prec	ARI
SC	0.239	0.056	0.299	0.179	0.010
DW	0.337	0.088	0.270	0.248	0.092
RTM	0.451	0.239	0.342	0.349	0.203
RMSC	0.295	0.139	0.320	0.204	0.049
TADW	0.455	0.291	0.414	0.312	0.228
GAE	0.408	0.176	0.372	0.418	0.124
VGAE	0.344	0.156	0.308	0.349	0.093
ARGA	0.573	0.350	0.546	0.573	0.341
ARVGA	0.544	0.261	0.529	0.549	0.245
MRGAE	0.627	0.361	0.587	0.601	0.363
GALA	**0.693**	**0.441**	–	–	**0.446**
DBGAN	0.670	0.407	–	–	0.414
MRGAE*	0.671	0.403	**0.620**	**0.620**	0.418

Table 12. Node clustering performance on Cora (left) and Citeseer (right).

3.2.2.2 Node Clustering. In this experiment, we consider another unsupervised task of clustering nodes in the network. We first compute the embeddings of Cora and Citeseer and perform K-means clustering on them, where K is set to be the number of node classes in each network. Then we follow the same procedure of previous work Pan et al. (2018); Shi, Fan, and Kwok (2019); Xia, Pan, Du, and Yin (2014) and match the predicted class labels with the ground-true labels using the Munkres assignment algorithm Munkres (1957). The results are evaluated based on accuracy (Acc), normalized mutual information (NMI), precision (Prec), F-score (F1) and average rand index (ARI).

Baselines Except for the baselines we use in the link prediction task, we include four more baseline algorithms that are designed for clustering: RTM Chang and Blei (2009), RMSC Xia et al. (2014), TADW C. Yang, Liu, Zhao, Sun, and Chang (2015), and GALA J. Park, Lee, Chang, Lee, and Choi (2019).

Results For a fair comparison, we first set the size of the output representations to 16-dim and record the results as MRGAE, and since GALA and DBGAN only report high dimensional performance, we then report the performance of our method with the same dimensions as DBGAN (i.e., 128-dim for Cora, 64-dim for Citeseer) and record the result as MRGAE*.

As shown in Table 12, our proposed method MRGAE outperforms the other methods on both datasets across all metrics. For the Cora dataset, the proposed method MRGAE* shows superior performance to GALA and DBGAN in almost all metrics except NMI. For the Citeseer dataset, MRGAE* and DBGAN perform similarly well while GALA gives the best results. It is mainly because GALA uses a 500-dim node representation which is much larger than DBGAN and MRGAE*.

3.2.2.3 Ablation Study. In this section, we validate the effectiveness of the multiview adversarial regularization module in our proposed method. We conduct the ablation experiments on both link prediction and node clustering tasks with the Cora dataset.

We first remove the view discriminator $\mathcal{D}_v$. By removing this part, the proposed method loses the ability to preserve the distribution consistency across the two specific views. We then remove the multiview reconstruction loss (MRL) and replace it with a simple GAE-based reconstruction loss. By removing this, the method losses rich information from the generated second view. Finally, we remove both parts. The three ablated methods are recorded as "w/o $\mathcal{D}_v$", "w/o MRL" and "w/o both", respectively.

As shown in Table 13, removing either part would cause a performance decrease on both link prediction and node clustering tasks, indicating the

Method	Link Prediction		Node Clustering				
	AUC	AP	Acc	NMI	F1	Prec	ARI
w/o $\mathcal{D}_v$	93.8	94.4	0.748	0.540	0.732	0.744	0.526
w/o MRL	94.0	94.1	0.706	0.527	0.686	0.712	0.496
w/o both	92.9	93.2	0.643	0.482	0.645	0.664	0.397
MRGAE	94.4	94.7	0.764	0.559	0.740	0.742	0.570

Table 13. Effectiveness evaluation of $\mathcal{D}_v$ and MRL.

effectiveness and necessity of $\mathcal{D}_v$ and MRL. The ablated method "w/o both" shows the biggest performance decrease, which consistently validates the claim.

3.2.2.4 30-day Unplanned ICU Patient Readmission Prediction.

In real-world networks, node content usually carries rich and important information for downstream applications, which highlights the practical value of the proposed method. Therefore, to better evaluate, we apply our method to a real-world application, i.e., unplanned ICU patient readmission prediction, to test if any performance gain can be achieved, compared with several baseline embedding methods.

We conduct this experiment based on Lin *et al.*'s work Lin et al. (2019), which leverages the embeddings of medical concepts (in the form of ICD-9 codes) in their method and achieves state-of-the-art performance. According to their claim, incorporating embeddings of medical concepts can benefit the prediction performance greatly. In this experiment, we test the 30-day unplanned ICU patient readmission prediction performance with different network embeddings for the ICD-9 ontology.

Dataset and Second View Construction In this experiment, we follow the data preprocessing procedure of previous work Harutyunyan et al. (2017); Lin et

al. (2019); Lu et al. (2019), and generate a dataset of 48,410 ICU stay records out of the freely available MIMIC-III database Johnson et al. (2016). The task is to predict whether or not a patient in an ICU stay will be readmitted within 30 days after discharge.

We take the transitive closure of ICD-9 as the first and main view. We first transform the short textual descriptions of nodes into one-hot representations, and compute the cosine similarities between them. If the cosine similarity between two nodes is greater than an empirical threshold of 0.7, we create a link between them in the second view of ICD-9.

Baselines Apart from the baselines used in the link prediction and node clustering task, we add one more strong baseline method, i.e., Poincaré Nickel and Kiela (2017), as the Poincaré method proves to be particularly effective in embedding hierarchical data, such as the ICD-9 ontology.

Experiment Settings We use the same metrics with Lin *et al.*'s work Lin et al. (2019). The *area under the Receiver Operating Characteristics curve* (AUC or A.R) is the main metric for evaluation. The recall rate of positive cases (Re-1), i.e., sensitivity, is also important in screening real patients. Additional metrics are reported, but they can be unstable and better be used for additional evaluation. Lin *et al.* use the embeddings for ICD-9 codes as part of their input. We replace the embeddings for ICD-9 with different methods.

Results As shown in Table 14, our proposed method achieves the best AUC score of 0.7807 with the highest sensitivity score of 0.7259. It is worth mentioning that the best reported AUC of Lin *et al.* is 0.791, but this is unfair to compare

Method	Acc	Pre-0	Pre-1	Re-0	**Re-1**	**A.R**	A.P
Poincaré	0.7223	0.9035	0.3740	0.7361	0.6655	0.7786	0.4827
GAE	0.7052	0.9061	0.3597	0.7089	0.6901	0.7712	0.4588
VGAE	0.7042	0.9007	0.3554	0.7127	0.6684	0.7653	0.4444
ARGA	0.7075	0.9126	0.3654	0.7057	0.7150	0.7757	0.4593
ARVGA	0.6966	0.9056	0.3518	0.6973	0.6934	0.7693	0.4519
MRGAE	0.7094	0.9157	0.3687	0.7055	**0.7259**	**0.7807**	0.4770

*Acc: Accuracy, Pre: Precision, Re: Recall, A.R: AUC under ROC, A.P: AUC under PRC

Table 14. Performance on 30-day unplanned ICU patient readmission prediction.

with since all the embeddings in the table are trained from the ICD-9 only, while the Claims embeddings Choi et al. (2016) they use are trained from millions of textual data.

3.2.3 Related Work. Recently, researchers use specifically designed encoders to aggregate the local neighborhood information of nodes, to generate low-dimensional embeddings. For example, GAE and VGAE Kipf and Welling (2016b) are two methods that use graph convolution networks (GCN) Kipf and Welling (2016a) as the encoder. VGAE uses the Gaussian distribution as a prior and pushes the learned representations close to this prior by incorporating a KL divergence penalty. ARGA and ARVGA incorporate an adversarial regularization framework for the same purpose, which is essentially similar in spirit to VGAE. Actually, incorporating adversarial regularization terms and matching the latent representations to a prior distribution is particularly useful for generating robust and meaningful representations when dealing with real-world complex graph data, which is first proposed by Adversarial Autoencoder (AAE) Makhzani et al. (2015). We extend the adversarial regularization framework to a multiview scenario where the distribution consistency across graph space and node content space is

to be preserved. But unlike previous methods like DBGAN Zheng et al. (2020) and DANE H. Gao and Huang (2018) which aim to reconstruct the node content directly, we focus on the semantic relatedness among them.

3.3 Predicting Patient Readmission Risk from Medical Text via Knowledge Graph Enhanced Multiview Graph Convolution

Patients who are readmitted to intensive care units (ICUs) after transfer or discharge usually have a greater chance of developing dangerous symptoms that can result in life-threatening situations. Readmissions also put families at higher financial burden and increase healthcare providers' costs. Therefore, it is beneficial for both patients and hospitals to identify patients that are inappropriately or prematurely discharged from ICU.

Over the past few years, there has been a surge of interest in applying machine learning techniques to clinical forecasting tasks, such as readmission prediction Lin et al. (2019), mortality prediction Harutyunyan et al. (2017), length of stay prediction Ma, Si, Wang, and Wang (2020), etc. Earlier studies generally select statistically significant features from patients' Electronic Health Records (EHRs), and feed them into traditional machine learning models like logistic regression Y. Xue et al. (2018). Deep learning models have also been gaining more and more attention in recent years, and have shown superior performance in medical prediction tasks. For example, Lin *et al.* select 17 types of chart events (diastolic blood pressure, capillary refill rate, etc.) over a 48-hour time window and put them into an LSTM-CNN model Lin et al. (2019) and achieve much better performance than previous work in readmission prediction.

A common theme among these studies is that they all rely on numerical and time-series features of patients, while neglecting rich information in the

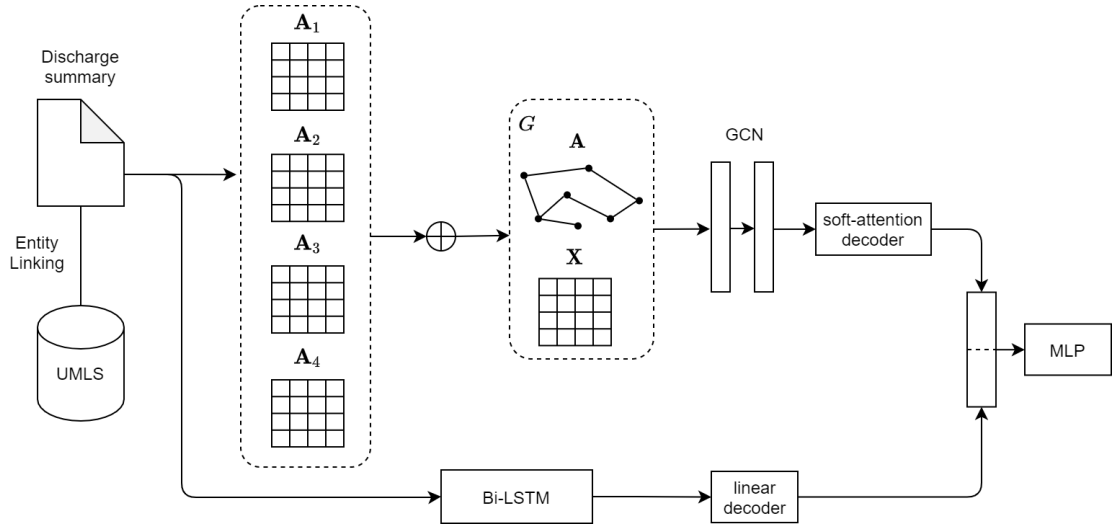


Figure 7. Architecture of MedText.

clinical notes of EHRs. This motivates us to tackle this task from a purely natural language processing perspective, which is not well explored in literature. Essentially, in this work, we consider the task of ICU readmission prediction as binary text classification, i.e., for a given clinical note, the model aims to predict whether or not the patient will be readmitted to ICU within 30 days after discharge.

Although it is possible to directly apply existing text classification methods to the readmission prediction task, two major challenges need to be addressed: (1) clinical notes, e.g., discharge summaries, are generally long and noisy, which makes it difficult to capture the inherent semantics to support classification; (2) general methods do not consider domain knowledge in the medical area, which is critical as medical concepts are hard to interpret with limited training for downstream tasks.

Recently, a useful strategy is proposed to tackle the first challenge, where it encodes documents with graphs-of-words to enhance the interactions of context, and to capture the global semantics of the document. The strategy has been

applied to different NLP tasks, including document-level relation extraction H. Chen, Hong, Han, Majumder, and Poria (2020); Christopoulou, Miwa, and Ananiadou (2019); Nan, Guo, Sekulić, and Lu (2020), question answering De Cao, Aziz, and Titov (2019); L. Qiu et al. (2019), and text classification Nikolentzos, Tixier, and Vazirgiannis (2020); Yao, Mao, and Luo (2019a); Y. Zhang et al. (2020). But constructing graphs of clinical notes for patient outcome prediction, to our knowledge, is underexplored.

Motivated by this, we propose a novel graph-based model that represents clinical notes as document-level graphs to predict patient readmission risk. Moreover, to address the second challenge, we incorporate an external knowledge graph, i.e., the Unified Medical Language System (UMLS) Bodenreider (2004) Metathesaurus, to construct a four-view graph for each input clinical note. The four views correspond to intra-document, intra-UMLS, and document-UMLS interactions, respectively. By constructing such an enhanced graph representation for clinical notes, we inject medical domain knowledge to improve representation learning for the model. Our contribution can thus be summarized as follows:

- We propose a novel graph-based text classification model, i.e., MedText, to predict ICU patient readmission risk from clinical notes in patients' EHRs. Unlike previous studies that rely on numerical and time-series features, we only use clinical notes to make predictions, which provides some insights on utilizing medical text for clinical predictive tasks.

- We construct a specifically designed multiview graph for each clinical note to capture the interactions among words and medical concepts. In this way, we inject domain-specific information from an external knowledge graph, i.e., UMLS, into the model. The experimental studies demonstrate the superb

performance of this method, by updating the state-of-the-art results on readmission prediction.

3.3.1 Method.

3.3.1.1 Graph Construction. For each document (e.g., clinical note), we construct a weighted and undirected four-view graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with an associated adjacency matrix $\mathbf{A}$, where $\mathcal{N}$ and $\mathcal{E}$ refer to the vertex set and edge set respectively. We also denote the representation of vertices by $\mathbf{X}$. Instead of using unique words in the document as vertices, we first conduct entity linking over the text and link the entity mentions to UMLS[1]. Consequently, we consider two types of vertices in the document-level graph $\mathcal{G}$, i.e., the unique words $\mathcal{N}_w$ and the linked UMLS entities $\mathcal{N}_e$. The vertex set $\mathcal{N}$ is thus formed as the union of $\mathcal{N}_w$ and $\mathcal{N}_e$: $\mathcal{N} = \mathcal{N}_w \cup \mathcal{N}_e$. Four views are then designed to exploit intra-document, intra-UMLS, and document-UMLS interactions that will be combined to form the adjacency matrix as follows.

Intra-Document: $\mathcal{V}_1$ $\mathcal{V}_1$ is designed to capture the intra-document interactions among words and entities. Essentially, we expect the edge weights between vertices to estimate the level of interaction, so that vertices can directly interact during message passing even if they are sequentially far away from each other in the document. In this work, we generate the adjacency matrix $\mathbf{A}_1$ for $\mathcal{V}_1$ by counting the co-occurrences of vertices within a fixed-size sliding window (size 3 in this work) over the text.

Intra-UMLS: $\mathcal{V}_2$,$\mathcal{V}_3$ In this work, we aim to inject external knowledge from UMLS to the document-level graph representation. To this end, we consider

[1]We use ScispaCy Neumann et al. (2019) as the entity linker in this work.

two types of information, i.e., the internal structure of UMLS and the semantic similarities between medical concepts. Specifically, we construct $\mathcal{V}_2$ by computing the shortest path lengths between entity vertices as edge weights in $\mathbf{A}_2$, where a shorter path indicates a higher relevance. We further construct $\mathcal{V}_3$ by computing the string similarities based on the word overlap ratios of entity descriptions for $\mathbf{A}_3$.

Document-UMLS: $\mathcal{V}_4$ $\mathcal{V}_4$ is constructed by calculating the cosine similarities between initial representations of all vertices, including words and entities, which aims to capture the interactions between the information sources, i.e., the document itself and the knowledge base. The similarities are used for edge weights $\mathbf{A}_4$.

View Combination By combining the four views, we expect to leverage three levels of interactions, i.e., intra-document, intra-UMLS, and document-UMLS, to generate rich interaction structures for documents to aid representation learning. Intuitively, the four views are combined via a weighted sum of the four adjacency matrices as the final adjacency matrix $\mathbf{A}$:

$$\mathbf{A} = \text{MASK}(\sum_{i=1}^{4} \alpha_i \mathbf{A}_i) \tag{3.22}$$

where $\mathbf{A}_i$ refer to each view's normalized adjacency matrix and α_i are the balancing factors that are determined by cross-validation. The adjacency matrix is then masked with a threshold, i.e., $\gamma = 0.5$, where only edges with larger weights are kept for further message passing. The motivation for the masking is to improve robustness and efficiency by decreasing some density.

The representation of vertices, i.e., $\mathbf{X}$, are initialized with a pre-trained word embedding BioWordVec Y. Zhang, Chen, Yang, Lin, and Lu (2019). For entity vertices, we take the average values of word embeddings of the entity names as the representation for the entity.

3.3.1.2 Encoding and Decoding. In this work, we incorporate a two-layer graph convolutional network (GCN) Kipf and Welling (2016a) to encode the graph representation of clinical notes, as depicted in Figure 7. We include an attention layer after GCN, which serves as a decoder to decode the document-level representation $\mathbf{D}_G$ from node embeddings. The encoding process can be described as:

$$\mathbf{X}^{(l+1)} = \text{LeakyReLU}(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{\frac{1}{2}}\mathbf{X}^{(l)}\mathbf{W}^{(l)}) \tag{3.23}$$

where $\hat{\mathbf{A}} = \mathbf{A}+\mathbf{I}$, and $\mathbf{I}$ is the identity matrix of $\mathbf{A}$. $\hat{\mathbf{D}}$ is the diagonal degree matrix of $\hat{\mathbf{A}}$, and $\mathbf{W}^{(l)}$ is the weight matrix for the l-th layer where $l = 0, 1, 2$ in this work.

We incorporate a graph summation module Y. Li, Tarlow, Brockschmidt, and Zemel (2015); Y. Zhang et al. (2020) to decode the document-level representation $\mathbf{D}_G$ from the constructed graph, by assigning different attention weights to the nodes. The decoding process can be described as:

$$\mathbf{X}_G = f_1(\mathbf{X}^{(2)}) \odot f_2(\mathbf{X}^{(2)}) \tag{3.24}$$

$$\mathbf{D}_G = \text{mean}(\mathbf{X}_G) + \max(\mathbf{X}_G) \tag{3.25}$$

where $\mathbf{X}^{(2)}$ is the output of the GCN encoder and f_1, f_2 are two feed-forward networks with sigmoid and leakyrelu activation, respectively. The f_1 network acts as a soft attention mechanism that indicates the relative importance of nodes, while f_2 serves as feature transformation. The operator $\odot$ denotes element-wise

multiplication. Then the document-level representation $\mathbf{D}_G$ is summarized as the addition of the mean and maximum values of the attentive node embeddings.

We also use a two-layer bidirectional LSTM to directly encode the document and decode the document-level representation $\mathbf{D}_T$ with a linear decoder, where linear transformation and max-pooling are applied. Then the two document-level representations, i.e., $\mathbf{D}_G$ and $\mathbf{D}_T$, are concatenated and fed into an MLP classifier. The model is optimized with cross-entropy loss.

3.3.2 Experiments.

Dataset The experiment is conducted based on the MIMIC-III Critical Care (Medical Information Mart for Intensive Care III) Database, which is a large, freely-available database composed of de-identified EHR data Johnson et al. (2016). For a fair comparison, we use the same data split with the baseline X. Zhang, Dou, and Wu (2020), where the discharge summaries are extracted from EHRs and the generated $48,393$ documents are split into training (80%), validation (10%), and testing (10%).

Evaluation Metrics We use three metrics for evaluation, i.e., the area under the receiver operating characteristics curve (AUROC), the area under the precision recall curve (AUPRC), and the recall at precision of 80% (RP80). AUROC and AUPRC are widely used for evaluating patient outcome prediction tasks, including readmission prediction Lin et al. (2019); Lu et al. (2019); X. Zhang et al. (2020). RP80 is a clinically-relevant metric that helps minimize the risk of alarm fatigue, as introduced in ClinicalBERT Huang et al. (2019), where we fix the precision at 80% and calculate the recall rate.

Baselines The following baselines are used for comparison.

- BioBERT. BioBERT is a domain-specific BERT variant pre-trained on large biomedical corpora, e.g., PubMed abstracts and PMC full-text articles Lee et al. (2020). In the experiment, we use the latest version, i.e., BioBERT v1.1, with a classification head as the baseline. The last 512 tokens of each note are used as input to the model.

- ClinicalBERT. ClinicalBERT is initialized from BioBERT v1.0 and pre-trained on MIMIC notes Alsentzer, Murphy, Boag, Weng, Jin, et al. (2019). Note that there is another ClinicalBERT Huang et al. (2019) model which presents a similar idea.

- CC-LSTM. Zhang *et al.* propose CC-LSTM that encodes UMLS knowledge into text representations and report state-of-the-art performance on readmission prediction on the MIMIC-III dataset X. Zhang et al. (2020). For a fair comparison, we use the same pre-trained word embeddings, i.e., BioWordVec Y. Zhang et al. (2019), in our model.

- MedText-x. Specifically, we replace the Bi-LSTM encoder with ClinicalBERT and BioBERT to demonstrate the effectiveness of the proposed graph-based knowledge injection strategy. The last two baselines are denoted by MedText-ClinicalBERT and MedText-BioBERT, respectively.

Results The experimental results are presented in Table 15. Generally, the proposed method, i.e., MedText, compares favorably with all the other baselines and outperforms the state-of-the-art method. Besides, directly applying pre-trained language models, such as BioBERT and ClinicalBERT, to readmission prediction

Method	AUROC	AUPRC	RP80
BioBERT	0.775	0.538	0.200
MedText-BioBERT	0.811	0.610	0.278
ClinicalBERT	0.781	0.536	0.189
MedText-ClinicalBERT	0.812	0.615	0.277
CC-LSTM X. Zhang et al. (2020)	0.804	0.613	N/A
MedText	**0.825**	**0.632**	**0.319**

Table 15. Performance on 30-day unplanned ICU patient readmission prediction.

Method	AUROC	AUPRC	RP80
w/o $\mathcal{V}_1$	0.803	0.605	0.300
w/o $\mathcal{V}_{1,2}$	0.809	0.615	0.296
w/o $\mathcal{V}_{1,2,3}$	0.801	0.607	0.290
w/o $\mathcal{V}_{1,2,3,4}$	0.799	0.601	0.288
w/o $\mathbf{D}_T$	0.808	0.601	0.275
Full	0.825	0.632	0.319

Table 16. Ablation analysis of MedText.

does not work well. It is most likely due to the long and noisy nature of clinical notes, and only the last 512 tokens are taken as input in the experiment. However, by combining with MedText, the performance gets improved greatly, indicating the effectiveness of the proposed graph-based knowledge injection method.

Additionally, Lin *et al.* propose a readmission prediction model that takes numerical features, e.g., chart events, of patients as input, and claims a state-of-the-art AUROC of 0.791 with AUPRC of 0.513 on the same dataset Lin et al. (2019). This is essentially not comparable as they are using numerical features instead of text, but it highlights the value of clinical notes in EHRs.

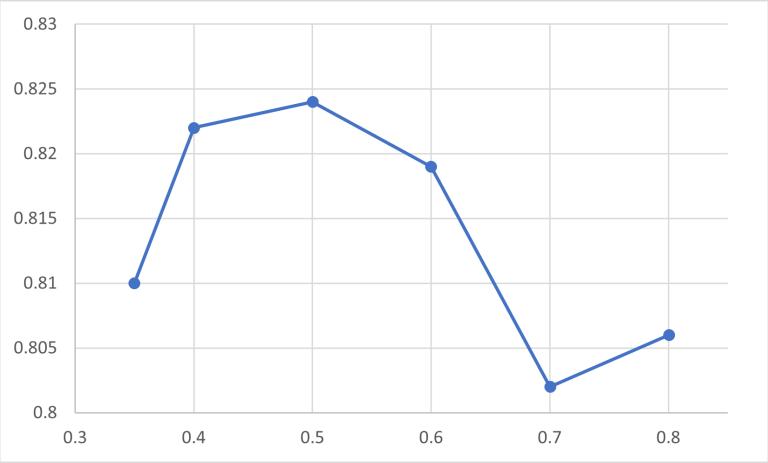


Figure 8. Sensitivity of masking threshold γ.

Ablation and Sensitivity Study We present the ablation study in Table 16. As shown in the table, removal of the four views will cause the performance to drop greatly, indicating the effectiveness and necessity of the four views. It is also worth noticing that the model still performs on par with CC-LSTM if the Bi-LSTM module is removed, i.e., w/o $\mathbf{D}_T$, and it would be more efficient in training. We also show the AUROC score with different masking thresholds in Figure 8, where AUROC reaches the peak when $\gamma = 0.5$. To further assess the performance of the model in terms of precision and recall, we show the P-R curve in Figure 9.

Error Analysis Entity linking plays an important role in this method as it is the first step of graph construction and all four views either directly or indirectly depend on the linked entities. Since a relatively high linking precision can be achieved by setting appropriate parameters of the ScispaCy linker, we mainly focus on the missed entities in the text. After manually examining a subset of notes, we roughly estimate that 15% to 25% of entities are not recognized or linked, which may have negatively influenced the prediction model. Some example snippets of clinical notes include:

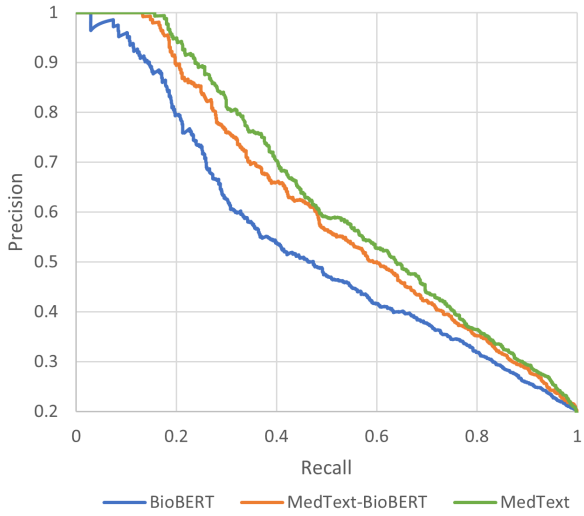


Figure 9. Precision-recall curve of MedText.

"this is a 69 year old man with a history of end stage cardiomyopathy (nyha class 4) and severe chf with an ef of 15 (ef of 20 on milrinone drip) as well as severe mr p/w sob , doe , pnd , weight gain of 6lbs in a week , likely due to chf **exacerbation** . "

"he has a history of v-tach which responded to amiodarone . patient also has icd in place . respiratory : sob and increased o2 requirement were likely secondary to chf **exacerbation** and resultant **pulmonary edema**"

"you were admitted for increasing **shortness of breath** and oxygen requirements on increasing doses of lasix"

The texts in bold refer to unrecognized entity mentions. Essentially they should be linked to UMLS entities C4086268 (Exacerbation), C0034063 (pulmonary edema) and C0013404 (Dyspnea), respectively. These uncovered entities might indicate the severity of patients' conditions and thus are critical for predicting the readmission risk.

3.3.3 Related Work. Earlier deep text classification models, such as TextCNN Y. Kim (2014) and TextRNN P. Liu, Qiu, and Huang (2016), mostly rely

on sequences of words for input representation. These methods focus on locality and lack the ability to capture long-distance and non-consecutive semantics Minaee et al. (2020); H. Peng et al. (2018); Y. Zhang et al. (2020), especially when the document is long. To mitigate this issue, some propose to encode documents with task-specific document-level graphs for representation learning H. Chen et al. (2020); Christopoulou et al. (2019); De Cao et al. (2019); Nan et al. (2020); L. Qiu et al. (2019); Yao et al. (2019a); Y. Zhang et al. (2020).

In this work, we represent documents as graphs of words and entities and encode them with a GCN-based model. The model is different from the aforementioned related work in that we construct a carefully designed four-view graph for each clinical note, and incorporate an external knowledge graph to enhance the graph representation. It is crucial to inject domain knowledge into prediction models as medical concepts are usually hard to capture with limited training for downstream tasks Lin et al. (2019); Lu et al. (2019); X. Zhang et al. (2020).

It is also worth noting that although large pre-trained language models, such as BioBERT and ClinicalBERT, have shown superior performance Alsentzer, Murphy, Boag, Weng, Jin, et al. (2019); Huang et al. (2019); Lee et al. (2020), applying them for long document classification remains an open problem, due to their quadratically increasing memory and time consumption Ding, Zhou, Yang, and Tang (2020). We consider it as a potential direction to take the full capacity of these models for text-based patient outcome prediction.

3.4 Conclusion

In concluding this chapter, we have examined three pioneering approaches for incorporating knowledge graphs into pre-trained language models and their

applications. Initially, we introduce a novel technique that harnesses medical notes within patients' EHR data, significantly improving state-of-the-art ICU readmission and in-hospital mortality prediction models. We specially leverage the hyperbolic embeddings of the ICD-9 ontology in our proposed method. To the best of our knowledge, we are the first to do so and achieve promising results.

Subsequently, we propose an innovative network embedding method, MRGAE, designed to maintain the consistency of node representations across two specific network views. This is achieved through the introduction of a multiview adversarial regularization module and a specially designed loss function for joint optimization. We conduct extensive and diverse experiments for evaluation, and the results demonstrate the superb performance of the proposed method.

Finally, we introduce MedText, a novel graph-based text classification model specifically designed to predict ICU patient readmission risk using clinical notes from patients' EHRs. The experiments demonstrate the effectiveness of the method and an updated state-of-the-art performance is observed on the benchmark.

Overall, these studies underscore the potential and effectiveness of leveraging knowledge graphs for enhancing the capabilities of pre-trained language models in the realm of healthcare. Moving forward to the next chapter, we pivot our attention to the other vital source of knowledge - clinical text, exploring the distinct strategies for its effective incorporation with language models.

CHAPTER IV

TEXT-BASED KNOWLEDGE INFUSION: STRATEGIES FOR DATA AUGMENTATION AND BEYOND

This chapter contains materials from the published papers "*Qiuhao Lu, Dejing Dou, and Thien Huu Nguyen. '**Textual Data Augmentation for Patient Outcomes Prediction.**' In Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 2817-2821. IEEE, 2021*", and "*Qiuhao Lu, Dejing Dou, and Thien Huu Nguyen. '**ClinicalT5: A Generative Language Model for Clinical Text.**' In Findings of the Association for Computational Linguistics: EMNLP 2022, pp. 5436-5443. 2022*". In these publications, the experiments were conducted solely by the author of the dissertation, Qiuhao Lu. Qiuhao also took complete responsibility for writing all the papers, and Thien Huu Nguyen contributed significantly by offering editorial feedback to enhance their quality.

In the previous chapter, we examine strategies for integrating domain knowledge into pre-trained language models and applications using knowledge graphs. In this chapter, we shift focus and present two innovative text-based knowledge infusion methods, leveraging data augmentation techniques.

Firstly, we introduce a novel approach for textual data augmentation to generate artificial clinical notes within patients' Electronic Health Records (EHRs). This serves as supplementary training data for patient outcomes prediction Lu, Dou, and Nguyen (2021b). Our method involves fine-tuning the generative language model GPT-2 to synthesize labeled text using the original training data. We propose a teacher-student framework, where a teacher model is initially pre-trained on the original data. Subsequently, a student model is trained on the GPT-

augmented data with guidance from the teacher model. We evaluate our approach on the widely studied patient outcome of 30-day readmission rates. Experimental results demonstrate that the augmented data enhances the predictive performance of deep models, emphasizing the effectiveness of our proposed architecture.

In the second study, we explore the potential of generative language models such as BART and T5, which have gained significant attention for their impressive performance in text generation and generative problem-solving tasks. However, the domain-specific variants of these models in the clinical domain have been relatively underexplored. To bridge this gap, we introduce ClinicalT5, a T5-based text-to-text transformer model pre-trained on clinical text Lu, Dou, and Nguyen (2022). We assess the proposed model intrinsically and extrinsically using various tasks and datasets. Our results demonstrate that ClinicalT5 outperforms T5 in domain-specific tasks and performs favorably when compared to closely related baseline models.

4.1 Textual Data Augmentation for Patient Outcomes Prediction

Patient outcomes, including patients' readmission risk, mortality rate, and length of stay (LOS), have been examined as important measurements for evaluating the quality of hospital care Davison et al. (2016). As the most commonly reported health outcome in the United States, readmissions are estimated to cost Medicare $15 billion annually, of which $12 billion is potentially preventable, according to the Medicare Payment Advisory Committee Hackbarth (2009). This highlights the importance of identifying patients at high risk of readmission.

Over the past few years, there has been a surge of interest in making predictions on patient outcomes using deep learning techniques, such as readmission

prediction Lin et al. (2019), mortality prediction Harutyunyan et al. (2017), length of stay prediction Ma et al. (2020), etc. Most of these studies heavily rely on feature engineering, where they select statistically significant features from patients' Electronic Health Records (EHRs), and feed them into deep models like a LSTM-CNN network Lin et al. (2019).

A common theme among these studies is that they all rely on numerical and time-series features of patients while neglecting the clinical notes of EHRs which prove to be informative in such predictive tasks. This motivates recent studies to cast this task as text classification, where the contextual content of EHRs is leveraged to make predictions. For example, Lu *et al.* propose a graph-based method that converts clinical notes to multi-view graphs and use them to predict ICU patients' 30-day unplanned readmission risk, surpassing state-of-the-art numerical-based methods Lu, Nguyen, and Dou (2021).

However, in real-world downstream applications, deep learning models often suffer from data limitations as they require large amounts of data for effective training. The situation is even worse in the biomedical domain due to the private and sensitive nature of this field. Despite data shortage, data imbalance is also an issue for patient outcomes prediction, e.g., only few patients are readmitted post-discharge. These data issues make patient outcomes prediction more challenging than general predictive tasks.

A natural solution to these problems is data augmentation, where new data is synthesized based on existing training data. This strategy has been actively applied in the field of computer vision, where researchers alter the training images to create a larger dataset by introducing random transformations such as translation, mirroring, rotation, and more McLaughlin, Del Rincon, and Miller

(2015). However, these augmentation strategies that are successful in computer vision cannot be easily applied to textual data due to the inherent complexity of natural language Amin-Nejad, Ive, and Velupillai (2020), where the grammatical or semantic consistency of text could hardly be preserved after transformation Anaby-Tavor et al. (2020). As to the specific task of readmission prediction, such issues, e.g., data imbalance, are either ignored Lu et al. (2019) or processed with sampling techniques Junqueira, Mirza, and Baig (2019), such as SMOTE Chawla, Bowyer, Hall, and Kegelmeyer (2002) or ROSE Menardi and Torelli (2014) that do not cope with textual data.

Recently, natural language generation (NLG) techniques have been leveraged as a new means for textual data augmentation. With the development of large pre-trained generative language models like GPT-2 Radford et al. (2019), researchers are able to generate high-quality and semantic-consistent textual data while preserving the annotated labels. This augmentation strategy has been applied in various NLP downstream tasks, such as event detection Veyseh, Lai, Dernoncourt, and Nguyen (2021), relation extraction Papanikolaou and Pierleoni (2020), commonsense reasoning Y. Yang et al. (2020), etc. However, in the biomedical field, leveraging GPT-2 to facilitate clinically-relevant predictive models is under-explored.

One main challenge of using GPT-2 for textual data augmentation is noise control. Existing studies typically address this issue in an isolated way, where they introduce heuristic filtering mechanisms to eliminate low-quality samples Anaby-Tavor et al. (2020) and feed the rest to the downstream model. However, such filtering strategies are prone to coverage errors and thus inevitably make incorrect judgments on the generated samples Veyseh et al. (2021), which would

cause false inclusion of good samples or false exclusion of bad samples. Moreover, the combined data samples are treated equally by the to-be-trained downstream model, and this would negatively impact the model as a consequence.

To overcome this issue, we propose a conceptually different strategy where all the generated samples are involved during training. We preserve all the generated samples in the first place and then introduce a teacher-student framework to regularize the representation learning of the generated samples with knowledge transferred from the original data. More specifically, we pre-train a teacher model on the original data and then train a student model on the combined data adaptively under the guidance of the teacher. The goal is to transfer the knowledge learned in the teacher model into the student model by enforcing a knowledge consistency between them, and that eventually the student model can be improved. We evaluate the framework with the state-of-the-art textual-based readmission prediction model Lu, Nguyen, and Dou (2021), the results of which indicate the effectiveness of the method.

The contributions of this work can be summarized as follows:

- We propose a novel architecture that leverages GPT-2 for **Med**ical text **Aug**mentation (MedAug) in the task of patient outcomes prediction. Essentially, we introduce a teacher-student framework that aims to control the noise of the generated text by enforcing knowledge consistency across the original and artificial texts.

- Taking the readmission prediction task as a case study, we specifically investigate the performance of MedAug with the state-of-the-art readmission prediction model as well as a baseline model. Extensive experiments

demonstrate that both models can improve their performance with the augmented data, indicating the effectiveness of the proposed architecture.

4.1.1 Method.

Notations In this study, we focus on textual-based readmission prediction models where the prediction task is cast as a supervised binary text classification problem. We refer to the original training dataset as $D_{train} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where x_i is a clinical note and $y_i \in \{0, 1\}$ indicates whether the patient is readmitted or not. Note that D_{train} is imbalanced where negative samples are 3x more than the positive ones, as only few patients are readmitted post-discharge. We similarly denote the test set by D_{test} and the validation set by D_{valid}. We also denote the synthesized training set by $D_{synthetic}$, which is generated by the fine-tuned GPT-2 model $\mathcal{G}_{tuned}$. We also combine the original and generated training data together to create a large training dataset $D_{combined} = D_{train} \cup D_{synthetic}$. Finally, we refer to the prediction method as $\mathcal{M}$.

Data Generation We fine-tune the GPT-2 model $\mathcal{G}$ on the original training data D_{train} so that it can synthesize reasonable textual data that can be used for the training of $\mathcal{M}$. To preserve the class information, we prepend the class label y_i to each note in the training data, i.e., $y_i\texttt{SEP}x_i\texttt{EOS}$, where `SEP` and `EOS` are the separation and ending token, respectively. We then fine-tune GPT-2 on the processed training data with the objective of predicting the next token, the same way it was pre-trained Radford et al. (2019). The fine-tuned model is regarded as $\mathcal{G}_{tuned}$.

For generating new data, we use the class label along with a short context as the prompt to G_{tuned}, i.e., $\texttt{prompt} = y_1\texttt{SEP}w_1w_2$ where the first two tokens

are included as context, as suggested in Anaby-Tavor et al. (2020). Since in our case the negative samples are 3x more than the positive ones, we only focus on generating positive samples to fulfill the gap, i.e., only the positive label y_1 is used for generation. We denote the generated training data by $D_{synthetic}$.

Data Integration As mentioned in the introduction, noise control is one of the main challenges for textual data augmentation. In this work, we propose a teacher-student framework for data integration so that all the generated samples are included for training. We first pre-train a teacher prediction model $\mathcal{M}_{teacher}$ on D_{train} to capture the inherent knowledge of the original clean training data. Then we train the student model $\mathcal{M}_{student}$ on the combined data $D_{combined}$ in a way that the teacher's knowledge can be used to guide the student learning. To achieve this, we aim to enforce knowledge consistency between the student and the teacher, by incorporating a KL divergence penalty to push the representations learned in the student model close to that in the teacher. Essentially, we seek to jointly minimize the KL divergence between the predicted label probability distribution of the student and the teacher, along with the original training objective of the student, i.e., $\mathcal{L} = \mathcal{L}_{student} + \tau\mathcal{L}_{KL}$. It's also worth mentioning that in this study we use the KL divergence to control noise in the labeled data generated by GPT-2, which is different from knowledge distillation on unlabeled data Hinton, Vinyals, and Dean (2015). The architecture is defined in Algorithm 1.

4.1.2 Experiments.

In this section, we evaluate the proposed framework on the task of ICU patients readmission prediction where we aim to show the effectiveness of MedAug. Essentially, we take as input the clinical note of patients' EHRs, and predict whether or not the patient will be readmitted within 30 days after discharge or transfer.

Algorithm 1: MedAug

Input: D_{train}, $\mathcal{G}$, $\mathcal{M}$
Output: $\mathcal{M}_{student}$

1 Fine-tune $\mathcal{G}$ on D_{train} to obtain $\mathcal{G}_{tuned}$
2 Use $\mathcal{G}_{tuned}$ to generate $D_{synthetic}$ and combine it with D_{train} to obtain $D_{combined}$
3 Pre-train a teacher model $\mathcal{M}_{teacher}$ on D_{train}
4 Train the student model $\mathcal{M}_{student}$ on $D_{combined}$ under the guidance of $\mathcal{M}_{teacher}$
5 **Return** $\mathcal{M}_{student}$

Dataset The experiment is conducted based on the MIMIC-III Critical Care (Medical Information Mart for Intensive Care III) Database, which is a large, freely-available database composed of de-identified EHR data Johnson et al. (2016). Following prior work X. Zhang et al. (2020), we extract the `Discharge Summaries` from EHRs as the data. For a fair comparison, we use the same data split with the baseline Lu, Nguyen, and Dou (2021) where $48,393$ generated documents are split into training (80%), validation (10%), and testing (10%). Specifically, the original training set D_{train} consists of 7555 positive samples and 30247 negative samples which are denoted by $D_{train,1}$ and $D_{train,0}$, respectively.

Evaluation Metrics We follow the prior work Lu, Nguyen, and Dou (2021) and use the area under the receiver operating characteristics curve (AUROC), the area under the precision-recall curve (AUPRC), and the recall at precision of 80% (RP80) for evaluation.

Prediction Models We consider the following two prediction models for evaluation in this experiment. We evaluate with two prediction models to investigate how MedAug performs when equipped with a base model and an advanced model.

– ClinicalBERT. ClinicalBERT is a domain-specific BERT variant initialized from BioBERT v1.0 Lee et al. (2020) and pre-trained on MIMIC notes Alsentzer, Murphy, Boag, Weng, Jin, et al. (2019). In this study, we add a linear classification head on top of it and use it as a baseline.

– MedText. MedText is a textual-based readmission prediction model and reports state-of-the-art performance on this task Lu, Nguyen, and Dou (2021).

Augmentation Baselines We consider two augmentation baselines for comparison.

– base. The base strategy is a baseline that all generated samples are included while no noise control is applied.

– LAMBADA. LAMBADA is an augmentation method specified for text classification Anaby-Tavor et al. (2020). Basically, they pre-train a classifier on the clean data and use it to select confident samples.

Results Table 17 shows the test performance of the two readmission prediction models, along with three augmentation strategies. We observe that without controlling the noise, i.e., base, both models demonstrate inferior performance, indicating the non-negligible level of noise in the generated samples. On the other hand, with MedAug, both models demonstrate better performance and the improvement is significant compared with the other two baselines, indicating the effectiveness of this framework.

4.1.3 Analysis. In this section, we investigate three potential issues that might have influenced the performance of MedAug, i.e., the number of

Method	AUROC	AUPRC	RP80
ClinicalBERT	0.782	0.549	0.201
ClinicalBERT-base	0.779	0.550	0.221
ClinicalBERT-LAMBADA	0.782	0.543	0.196
ClinicalBERT-MedAug	**0.791**	**0.565**	**0.234**
MedText	**0.823**	0.632	0.319
MedText-base	0.803	0.599	0.290
MedText-LAMBADA	0.806	0.604	0.266
MedText-MedAug	0.822	**0.633**	**0.328**

Table 17. Test performance on 30-day unplanned ICU patient readmission prediction.

$\|D_{synthetic}\|$	Method	AUROC	AUPRC	RP80
3k	ClinicalBERT	0.777	0.550	0.220
9k	ClinicalBERT	0.784	0.567	0.246
12k	ClinicalBERT	0.784	0.569	0.245
24k	ClinicalBERT	0.783	0.566	0.251
3k	MedText	0.812	0.621	0.329
9k	MedText	0.811	0.623	0.337
12k	MedText	0.806	0.611	0.311
24k	MedText	0.809	0.618	0.331

Table 18. Influence of $|D_{synthetic}|$ by MedAug.

synthesized samples $|D_{synthetic}|$, the fine-tuning and generation strategy for GPT-2, and the version of GPT-2.

Number of Synthesized Samples Table 18 shows the validation performance of different $|D_{synthetic}|$, demonstrating the influence of the size of the synthetic training set. With the increase of synthesized samples, the general performance appears to have reached a peak and then begin to drop slightly. We conjecture

Prompt	Balanced	Method	AUROC	AUPRC	RP80
w/o ctx	N	ClinicalBERT	0.771	0.535	0.205
w/o ctx	Y	ClinicalBERT	0.773	0.536	0.216
w/ ctx	N	ClinicalBERT	0.767	0.531	0.198
w/ ctx	Y	ClinicalBERT	0.775	0.551	0.226
w/o ctx	N	MedText	0.791	0.589	0.296
w/o ctx	Y	MedText	0.791	0.595	0.313
w/ ctx	N	MedText	0.791	0.593	0.296
w/ ctx	Y	MedText	0.795	0.602	0.318

Table 19. Influence of GPT-2 fine-tuning/generation strategies.

that there is a trade-off between size and performance, and it is determined by the augmentation strategy.

GPT-2 Fine-tuning Strategy It is common that patient outcomes demonstrate an imbalanced distribution, e.g., only few patients are readmitted after discharge. In our case, negative samples are 3x more than the positive ones, i.e., $D_{train,0} = 4 \times D_{train,1}$. Therefore, when fine-tuning GPT-2 using the original training data, we explicitly make it balanced to prevent the negative samples from misleading GPT-2, by performing random under-sampling over D_{train}. As to the prompt to GPT-2 in generating new samples, we compare two options, i.e., w/ and w/o context, where context refers to the first two tokens of the text.

We investigate the two issues and show the comparison results on the validation set in Table 19. Note that to avoid the impact from augmentation strategies, we use the base method, i.e., simply include all the samples, in this experiment. Generally, a balanced training set and a prompt with context are the best options for fine-tuning and generation with GPT-2 in this task.

GPT-2 version	Method	AUROC	AUPRC	RP80
small	ClinicalBERT	0.784	0.567	0.246
medium	ClinicalBERT	0.783	0.568	0.252
small	MedText	0.811	0.623	0.337
medium	MedText	0.811	0.623	0.339

Table 20. Influence of the version of GPT-2.

GPT-2 Version Finally, we investigate the version of GPT-2 and its influence over the quality of synthesized samples. We test with GPT-2-small and GPT-2-medium and show the results in Table 20. Generally, we observe that GPT-2-medium has a minor advantage over GPT-2-small. However, considering the training cost and efficiency, we choose to use GPT-2-small for all the experiments in this study.

4.1.4 Related Work.

Readmission prediction is a challenging task and has attracted a lot of attention over the years. Lin *et al.* select numerical chart event features over a 48-hour time window and feed them to a deep LSTM-CNN network Lin et al. (2019) and achieve much better performance than traditional methods. Zhang *et al.* propose CC-LSTM that encodes external knowledge into text representations and outperforms Lin's work X. Zhang et al. (2020). Afterward, Lu *et al.* propose to convert clinical notes to multi-view graphs and process them with graph convolution networks Lu, Nguyen, and Dou (2021). These studies demonstrate the value of textual content in EHRs and motivate us to apply textual data augmentation to this task.

Recently, using GPT-2 for augmenting textual training data has been studied for a variety of tasks in the NLP field, such as event detection Veyseh et al. (2021), relation extraction Papanikolaou and Pierleoni (2020), commonsense

reasoning Y. Yang et al. (2020), spoken language understanding B. Peng, Zhu, Zeng, and Gao (2020), extreme multi-label classification D. Zhang, Li, Zhang, and Yin (2020), etc. However, none of these works has leveraged GPT-2 for patient outcomes prediction. This highlights the importance of this study and motivates us to explore more of this direction.

4.2 ClinicalT5: A Generative Language Model for Clinical Text

In the past few years, large pre-trained language models (PLMs), such as BERT Devlin et al. (2019), RoBERTa Y. Liu et al. (2019), GPT-3 Brown et al. (2020), BART M. Lewis et al. (2020), T5 Raffel et al. (2020), etc., have achieved great success over a variety of downstream tasks in natural language processing (NLP). These PLMs mainly depend on self-supervised pre-training on large amounts of general-domain textual data, e.g., Wikipedia, news articles, web crawl corpus, etc., and are widely adopted in downstream applications. Despite the superior performance of these PLMs on general-domain text, their performance over domain-specific text is relatively poor Ma et al. (2019). To bridge this gap, researchers propose to build domain-specific PLMs through fine-tuning or pre-training from scratch over domain corpora. For example, in the biomedical and clinical domains, various domain-specific PLMs have been explored and released, including BioBERT Lee et al. (2020), SciBERT Beltagy et al. (2019), BlueBERT Y. Peng, Yan, and Lu (2019a), ClinicalBERT Huang et al. (2019), BioClinicalBERT[1] Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019), umlsBERT Michalopoulos et al. (2020), diseaseBERT Y. He et al. (2020a), SciFive Phan et al. (2021), and BioBART H. Yuan et al. (2022).

[1] Also known as ClinicalBERT.

Domain-specific language models have been extensively explored in different kinds of NLP-related downstream applications, ranging from entity linking Bhowmik, Stratos, and de Melo (2021) to document classification Allada et al. (2021). Generally, a typical and popular usage of the aforementioned PLMs is to leverage them to encode domain text, the learned representations of which are then fed into some task-specific structures for label prediction. Taking a complicated real-world task as an example, Huang et al. (2019) predicts patients' risk of readmission within 30 days after discharge using clinical notes in the Electronic Health Records (EHRs). Essentially, they encode discharge summaries of patients with ClinicalBERT, and put the learned embeddings of the `[CLS]` token to a linear layer on top for prediction, leading to better performance than traditional models. Moreover, Lu, Nguyen, and Dou (2021) constructs a document-level multi-view graph out of each clinical note and predicts patients' 30-day readmission risk with a graph-based model, and they use BioClinicalBERT Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019) as the encoder within the graph model.

Recently, generative language models, e.g., BART M. Lewis et al. (2020) and T5 Raffel et al. (2020), have attracted attention since they are naturally effective for natural language generation tasks, such as document summarization J. Chen and Yang (2021), question answering Sachan et al. (2021); Zhu et al. (2021), data augmentation Lu, Dou, and Nguyen (2021b), etc. Meanwhile, a novel paradigm of leveraging generative language models has gained popularity, where researchers cast non-generation tasks as generative problems, e.g., to directly generate textual labels to incorporate their semantics, and report promising results De Cao, Izacard, Riedel, and Petroni (2021); De Cao et al. (2022). However, such approaches are still underexplored in certain domains due to lack of domain-

specific generative language models, i.e., most of the aforementioned domain-specific PLMs are notably domain-adapted BERT-style models. In the biomedical domain, two generative language models SciFive Phan et al. (2021) and BioBART H. Yuan et al. (2022) have been released, but in the clinical domain, the situation is worse and no such generative models exist to our knowledge. Though the two domains are relatively close, clinical text poses unique challenges compared to general and non-clinical biomedical text due to its specific linguistic characteristics Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019). Previous studies list some of the linguistic features of clinical text, e.g., heavy use of professional technical terminology, abbreviations and acronyms, passive verbs, omission of subjects and verbs, etc., and these features make clinical text divergent from standard language Smith et al. (2014).

Aiming to fulfill this gap, we adapt T5 Raffel et al. (2020) to the clinical domain by training a domain-specific variant using clinical text, i.e., ClinicalT5. We demonstrate the capabilities of the model by conducting both intrinsic and extrinsic evaluations. For intrinsic evaluation, we aim to evaluate its capability to capture the similarity and relatedness of the Unified Medical Language System (UMLS) concept pairs, where we measure the correlation coefficient between the similarity scores of the encoded representations for the concept pairs and those judged by human experts. For extrinsic evaluation, we evaluate the proposed model along with baselines over a diverse set of benchmark datasets, ranging from document classification (DC), named entity recognition (NER), to natural language inference (NLI). Furthermore, we also evaluate on three more complicated real-world tasks of clinical importance, i.e., patients' 30-day readmission risk, 30-day

and 1-year mortality risk. We show that ClinicalT5 dramatically outperforms T5 and compares favorably with its close baselines across all of these tasks.

4.2.1 Related Work.

Biomedical Domain-Adapted Models The biomedical domain has been an active area of research in the NLP community for the past few years. Many relevant studies have been presented, ranging from domain-specific language models, external knowledge infusion, and various downstream applications, etc. Beltagy et al. (2019); Y. He et al. (2020a); Lee et al. (2020); Lu, Dou, and Nguyen (2021a); Michalopoulos et al. (2020); Y. Peng et al. (2019a). Most of the biomedical language models are BERT Devlin et al. (2019) variants fine-tuned to biomedical text, e.g., BioBERT is trained on PubMed abstracts and PMC full text articles Lee et al. (2020) and SciBERT is trained on the full text of biomedical and computer science papers from the Semantic Scholar corpus Beltagy et al. (2019). Besides, researchers inject external domain knowledge into adapted biomedical language models due to the knowledge-intensive nature of this domain, e.g., umlsBERT is directly trained using UMLS text Michalopoulos et al. (2020), He *et al.* infuse disease information from the corresponding Wikipedia passages into language models Y. He et al. (2020a), and Lu *et al.* inject biomedical knowledge from multiple sources into language models via adapters Lu, Dou, and Nguyen (2021a). For generative language models, SciFive is an adapted T5 model pre-trained on PubMed abstracts and PMC articles Phan et al. (2021) and BioBART is an adapted BART model pre-trained on PubMed abstracts H. Yuan et al. (2022).

Clinical Domain-Adapted Models In the clinical domain, there are mainly two popular BERT models, i.e., ClinicalBERT Huang et al. (2019) and

BioClinicalBERT Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019), which are both trained on the clinical notes in the MIMIC-III database Johnson et al. (2016). For generative language models, however, the topic is not well explored and this situation motivates our work.

4.2.2 ClinicalT5. Following prior studies on clinical language models Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019); Huang et al. (2019), we use the textual notes in MIMIC-III to train ClinicalT5, which consists of approximately 2 million notes. Similarly, only minimal pre-processing is conducted where unnecessary tokens and characters are removed Huang et al. (2019).

In particular, we initialize the weights from the SciFive-PubMed-PMC model (base and large) Phan et al. (2021) and further pre-train with the span-mask denoising objective Raffel et al. (2020) on the pre-processed MIMIC-III notes. The base and large models have $\sim 220M$ parameters with 12 layers and $\sim 770M$ parameters with 24 layers, respectively. For each of the two versions, we further pre-train ClinicalT5 on the unlabeled text for extra $10k$ steps, with a max sequence length of 512, a batch size of 8, and a learning rate of 1e$-$4. The pre-training is performed on 3 Nvidia Tesla V100-32GB GPUs. We refer the readers to Raffel et al. (2020) for a more detailed treatment of the architecture and training objectives of T5.

4.2.3 Experiments. In this section, we evaluate ClinicalT5 both intrinsically and extrinsically, along with the following generative baselines (for both general and domain-specific texts): BART M. Lewis et al. (2020), BioBART H. Yuan et al. (2022), T5 Raffel et al. (2020), SciFive Phan et al. (2021), to demonstrate the capabilities of ClinicalT5 across different applications.

Model	UMNSRS-Similarity		UMNSRS-Relatedness	
	Pearson	Spearman	Pearson	Spearman
BART-base	0.1456	0.1300	0.0756	0.0625
BioBART-base	**0.3753**	**0.3441**	**0.3101**	**0.2929**
T5-base	0.2050	0.1448	0.1056	0.0519
SciFive-base	0.1941	0.1488	0.1359	0.0900
ClinicalT5-base	0.2126	0.1611	0.1478	0.0948
BART-large	0.2234	0.1958	0.1706	0.1546
BioBART-large	**0.4511**	**0.4302**	**0.3517**	**0.3400**
T5-large	0.2379	0.2018	0.1813	0.1564
SciFive-large	0.3176	0.2642	0.3039	0.2618
ClinicalT5-large	0.3391	0.2847	0.2884	0.2468

Table 21. Pearson's and Spearman's correlation coefficient scores.

4.2.3.1 Intrinsic Evaluation. We conduct intrinsic evaluation on the datasets UMNSRS-Sim and UMNSRS-Rel Pakhomov et al. (2010), which consist of 566 and 587 UMLS term pairs respectively. Each pair comes with a *similarity* score and a *relatedness* score that are manually assigned by human experts. Similar to previous work Y. Zhang et al. (2019), we encode the terms with ClinicalT5 and the baselines. Essentially, we use the mean-pooled vectors of the last hidden states of the encoders as the term embeddings and calculate a cosine similarity score for each pair. Then we compute the Pearson's correlation coefficient and Spearman's correlation coefficient between the computed scores and the expert-assigned scores. As shown in Table 21, ClinicalT5 demonstrates a better ability to capture the similarity of UMLS terms than T5 and Scifive, indicating the effectiveness of the training.

4.2.3.2 Extrinsic Evaluation. For extrinsic evaluation, we consider three different tasks, i.e., document classification (DC), named entity recognition (NER), and natural language inference (NLI). To validate the models' capability

Tasks	HOC			NCBI			BC5CDR			MEDNLI
Metrics(%)	P	R	F1	P	R	F1	P	R	F1	Acc
BART-base	80.30	79.84	79.81	62.23	72.09	66.80	59.24	67.26	63.00	75.60
BioBART-base	84.68	83.54	83.82	63.10	71.77	67.16	61.78	72.05	66.52	80.66
T5-base	82.00	80.98	81.19	86.64	83.00	84.78	80.73	81.68	81.20	81.86
SciFive-base	85.10	84.83	84.70	86.43	88.25	87.33	**83.56**	81.43	**82.48**	83.90
ClinicalT5-base	**85.44**	**85.14**	**85.06**	**87.28**	**88.56**	**87.92**	81.55	**82.92**	82.23	**84.95**
BART-large	84.89	84.07	84.18	63.39	74.50	68.50	66.45	62.07	64.19	84.53
BioBART-large	84.80	84.51	84.39	67.74	70.51	69.10	65.00	71.93	68.29	**86.29**
T5-large	85.42	84.75	84.79	84.20	84.99	84.60	78.31	79.75	79.02	83.83
SciFive-large	**85.57**	**85.67**	**85.34**	85.91	85.10	85.50	78.28	79.89	79.08	84.95
ClinicalT5-large	85.37	84.79	84.78	**86.37**	**87.09**	**86.73**	**79.24**	**81.49**	**80.35**	85.86

Table 22. Performance comparison over document classification, named entity recognition, and medical natural language inference.

on clinical text, we select datasets that are closely relevant to clinical targets rather than biomedical or chemical related data such as BC5CDR-chemical J. Li et al. (2016). We fine-tune the evaluating models on 4 corresponding datasets across these tasks in a single-task text-to-text manner. For all the experiments, we use a batch size of 16 and a learning rate of 1e−4. Due to different targets, we set the max source text length to 256, and the max target text lengths to 52, 256, 256, 15 for the datasets HOC, NCBI, BC5CDR and MEDNLI, respectively.

Document Classification We conduct document classification on the HOC dataset Baker et al. (2016), which consists of $9,972$ samples for training and $4,947$ samples for testing. Essentially, we fine-tune the evaluating models to categorize the texts into 10 categories by directly generating the class labels, e.g., "`empty`", "`evading growth suppressors`", "`genomic instability and mutation`", etc.

Named Entity Recognition We conduct named entity recognition on two popular datasets, i.e., NCBI-disease Doğan, Leaman, and Lu (2014) and BC5CDR-disease J. Li et al. (2016). The input text sequence may contain a disease term

and the term should be identified and labeled in the target text, e.g., for the input text "Genotype and phenotype in patients with dihydropyrimidine dehydrogenase deficiency", the target is "Genotype and phenotype in patients with `disease*` dihydropyrimidine dehydrogenase deficiency `*disease`".

Natural Language Inference We conduct natural language inference evaluation on the MEDNLI dataset Romanov and Shivade (2018b), which consists of $11,232$ training samples and $1,422$ testing samples. Essentially, we convert the premise-hypothesis pair to a sequence and prepend a task-specific prefix to it, e.g., "mednli: premise: [...]. hypothesis: [...]." We take the converted sequence as the input text and fine-tune the evaluating models to generate the target labels, i.e., "`contradiction`", "`neutral`", "`entailment`".

Results The results are shown in Table 22. Generally, ClinicalT5 outperforms T5 and SciFive across most of these metrics, and the advantage indicates the success of the training over clinical text. However, ClinicalT5-large is on par with T5-large and has a slightly lower recall than SciFive-large on the HOC dataset. We conjecture that the large versions of BART and T5 already have enough capacity for the task which makes domain-specific training less impressive, as reflected by the fact that BioBART-large is only marginally better than BART-large. For MEDNLI, ClinicalT5 consistently outperforms T5 and SciFive although BioBART-large achieves the highest accuracy.

4.2.3.3 Real-world Evaluation. We also evaluate the models on more complicated real-world applications of clinical importance, i.e., 30-day unplanned ICU patient readmission risk, 30-day and 1-year patient mortality risk. The experiment is conducted based on the MIMIC-III dataset Johnson et

Tasks	30-d Readmission			30-d Mortality		1-y Mortality	
Metrics(%)	A.R.	A.P.	RP80	A.R.	A.P.	A.R.	A.P.
T5-base	77.10	52.24	16.97	80.03	23.62	78.52	45.72
SciFive-base	**78.12**	53.95	18.87	80.38	24.16	78.95	45.38
ClinicalT5-base	77.94	**54.25**	**19.76**	**81.11**	**26.70**	**79.09**	**46.58**

A.R: AUC under ROC, A.P: AUC under PRC, RP80: recall at precision of 80%

Table 23. Performance on patients' outcomes prediction.

al. (2016). Following previous work Lu, Nguyen, and Dou (2021); X. Zhang et al. (2020), we extract the discharge summaries from EHRs and generate 48,393 documents. Essentially, we take the evaluating models to encode the last 512 tokens of each note, the last hidden states of which are fed into a linear layer on top for prediction. As shown in Table 23, ClinicalT5 shows the best results across almost all the metrics, demonstrating its potential for real-world applications in the clinical domain.

4.2.4 Limitations. In this work, we present a generative language model for clinical texts based on T5. Although our experiments demonstrate the effectiveness of our method, there are still some limitations that can be improved in future work. First, our evaluation has not included question answering and other related tasks for clinical texts. These are important tasks Phan et al. (2021) and can be further explored in future work. Second, our pre-training method for ClinicalT5 has mainly inherited the objectives from T5 using direct unlabeled texts. As such, many important domain-specific knowledge for the clinical domain (e.g., knowledge bases, concept definition) has not been explored to improve our generative model, serving as a promising direction for future research.

4.2.5 Ethics Statement. All datasets used in this research are publicly available and are obtained according to each dataset's respective data usage policy. We avoid showing any direct excerpts of the data in the paper. We

do not attempt to identify or deanonymize users in the data in any way during our research, thus preventing any bias in our methods toward any specific users.

More specifically, the proposed models are trained on the clinical notes of the public MIMIC-III database, which are already deidentified in accordance with Health Insurance Portability and Accountability Act (HIPAA) standards using structured data cleansing and date shifting. As such, all identifying data elements in HIPAA, including patient name, telephone number, address, and dates, are already removed Johnson et al. (2016) from our training data to hinder attempts to retrieve personal information from our models. Similar to existing pre-trained and publicly available models for the clinical domain, i.e., ClinicalBERT Huang et al. (2019) and BioClinicalBERT Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019), the proposed models serve as a resource to facilitate future research on clinical text.

4.3 Conclusion

In this chapter, we have examined two innovative strategies for integrating clinical text as a source of domain knowledge into pre-trained language models. Firstly, we introduce MedAug, a framework leveraging the power of GPT-2 to create artificial training data for patient outcome prediction. We evaluate the method on the task of ICU patients readmission prediction, the results of which demonstrate that either a baseline or an advanced prediction model can benefit from the synthesized training data, under the framework of MedAug. Essentially, to control the noise in the synthesized data, we propose a teacher-student architecture that enforces knowledge consistency across the original and artificial texts. We introduce a mechanism for knowledge consistency enforcement to mitigate noises from generated data based on KL divergence. While the improvement in advanced

models is less pronounced than in baseline models, this preliminary exploration provides a foundation for further study.

Next, we explore and propose ClinicalT5, a clinical text-focused variant of the T5-based text-to-text transformer model. We evaluate the proposed model both intrinsically and extrinsically, and the results show that ClinicalT5 compares favorably with its close baselines. Further testing on more complex patient outcome prediction tasks demonstrates its potential for real-world downstream tasks in the clinical domain.

These investigations highlight the importance and potential of harnessing clinical text as a source of domain knowledge in enhancing pre-trained language models. As we move to the next chapter, we introduce an innovative, parameter-efficient approach for infusing knowledge from diverse sources and formats into pre-trained language models, thereby broadening their capabilities in domain-specific tasks.

CHAPTER V

DOMAIN ADAPTATION WITH ADAPTERS: PARAMETER-EFFICIENT APPROACHES TO KNOWLEDGE INCORPORATION

This chapter contains materials from the published paper "*Qiuhao Lu, Dejing Dou, and Thien Huu Nguyen. '**Parameter-efficient domain knowledge integration from multiple sources for biomedical pre-trained language models.**' In Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 3855-3865. 2021*". In this publication, the experiments were conducted solely by the author of the dissertation, Qiuhao Lu. Qiuhao also took complete responsibility for writing the paper, and Thien Huu Nguyen contributed significantly by offering editorial feedback to enhance its quality.

This chapter investigates the integration of domain knowledge into PLMs using adapters as a parameter-efficient approach to enhance their performance in clinical settings.

In particular, we present an architecture specifically designed to efficiently incorporate domain knowledge from diverse sources into PLMs Lu, Dou, and Nguyen (2021a). Our approach utilizes adapters, which are small bottleneck feed-forward networks inserted between intermediate transformer layers in PLMs, to encode domain-specific knowledge. These knowledge adapters are pre-trained for individual domain knowledge sources and combined using an attention-based knowledge controller to enrich PLMs. In the context of the biomedical domain, we explore three knowledge-specific adapters based on the UMLS Metathesaurus graph, Wikipedia articles on diseases, and semantic grouping information for biomedical concepts. Through extensive experiments conducted on various biomedical Natural Language Processing (NLP) tasks and datasets, we demonstrate

the advantages of the proposed architecture and knowledge-specific adapters across multiple PLMs.

The proposed adapter-based approach offers a significant advantage over traditional full-model fine-tuning, which requires substantial computational resources and can sometimes lead to "catastrophic forgetting" where the model may overwrite previously learned general-domain knowledge during the fine-tuning process. By utilizing adapters, we can efficiently and selectively modify parts of the pre-trained models to encode the domain-specific knowledge, thus requiring fewer parameters and less computational power. This approach provides an efficient, scalable, and effective method for knowledge integration, making it a promising technique for improving the performance of clinical NLP tasks. With the use of adapters, we can ensure the models remain adaptable and robust while maintaining their original general-domain knowledge, thereby promoting a more efficient and effective application of PLMs in the clinical domain.

5.1 Parameter-Efficient Domain Knowledge Integration from Multiple Sources for Biomedical Pre-trained Language Models

In the past few years, large pre-trained language models (PLMs) have demonstrated superior performance over various downstream tasks in natural language processing (NLP), such as BERT Devlin et al. (2019), RoBERTa Y. Liu et al. (2019), ALBERT Lan et al. (2019), GPT-3 Brown et al. (2020), etc. These PLMs mainly depend on self-supervised pre-training on large amounts of textual data, e.g., Wikipedia, and can be conveniently applied to downstream tasks via fine-tuning. Despite the great success of these general PLMs, their performance over domain-specific texts is relatively poor due to domain shifts Ma et al. (2019). Consequently, recent studies construct domain-specific PLMs through fine-tuning or

pre-training from scratch over domain corpora, such as BioBERT Lee et al. (2020), ClinicalBERT Huang et al. (2019), SciBERT Beltagy et al. (2019), etc.

Since these PLMs are mostly pre-trained on unstructured free texts, a common issue among the aforementioned general and domain-specific PLMs is their lack of specific structured knowledge, which results in their incompetence on knowledge-driven tasks Rogers, Kovaleva, and Rumshisky (2020). For instance, some studies point out PLMs are insufficient to well capture factual knowledge from text Poerner, Waltinger, and Schütze (2019); R. Wang et al. (2020); X. Wang et al. (2021).

To enrich PLMs with external knowledge, some efforts have been made recently B. Kim et al. (2020); Levine et al. (2020); X. Wang et al. (2021); Yao et al. (2019b); Z. Zhang et al. (2019). A common theme among these approaches is the incorporation of an auxiliary knowledge-driven training objective. For instance, KG-BERT Yao et al. (2019b) integrates world/factual knowledge from Wikipedia via knowledge graph completion; KEPLER X. Wang et al. (2021) introduces a Knowledge Embedding objective and combines it with the language modeling objective for joint optimization. Despite the improved performance of these knowledge-enriched PLMs over downstream tasks, there are three limitations. First, these approaches, either training from scratch or fine-tuning over off-the-shelf checkpoints, need to optimize the entire model, which is quite expensive. Second, they mostly focus on single-source knowledge incorporation, e.g., an encyclopedia, and neglect knowledge from multiple sources. This limits the utilization of potential knowledge, especially for knowledge-sensitive areas such as the biomedical domain where knowledge is stored in multiple sources and formats Jin, Dhingra, Cohen, and Lu (2019); Lee et al. (2020). Third, most of the existing knowledge integration

approaches focus on general domain knowledge, while domain knowledge infusion for PLMs is underexplored.

To address these limitations, we propose to perform knowledge integration for PLMs via adapters Houlsby et al. (2019); Pfeiffer, Kamath, Rücklé, Cho, and Gurevych (2021); Pfeiffer et al. (2020); Rebuffi, Bilen, and Vedaldi (2017); R. Wang et al. (2020). Basically, adapters are lightweight neural networks that are placed inside PLMs. When fine-tuning a PLM, the original parameters of the PLM are fixed and only the adapters are fine-tuned. This makes adapters a parameter-efficient alternative to full model fine-tuning. Another benefit of adapters is their independent nature, where multiple adapters can be trained independently without interfering with each other. As such, we propose to enrich PLMs with adapters that are independently pre-trained for different sources of domain knowledge.

In this paper, we propose an architecture that aims to integrate domain knowledge from multiple sources via knowledge-specific adapters to enrich PLMs. We take the biomedical domain as a case study, as it is a knowledge-sensitive area where domain knowledge is essential for various NLP applications. Specifically, we explore three knowledge-specific adapters for PLMs based on the UMLS Metathesaurus graph, the Wikipedia articles for diseases, and the semantic grouping information for biomedical concepts. We also incorporate an attention-based knowledge controller module that aims to adaptively adjust the activation levels of the adapters, which also brings some explainability as it shows the importance of the adapters for a task. The experimental results show that by equipping PLMs with domain knowledge from multiple sources via the proposed architecture, their overall performance gets consistently improved across tasks

and datasets. Moreover, the pre-trained adapters can be directly integrated with multiple PLMs, demonstrating transferability of the architecture.

The contributions of this work can be summarized as follows:

- We propose a novel architecture that incorporates **D**iverse **A**dapters for **K**nowledge **I**ntegration (DAKI) into PLMs. It integrates domain knowledge from multiple sources adaptively via an attention-based knowledge controller. The architecture demonstrates effectiveness, transferability, explainability as well as parameter-efficiency in experiments.

- Taking the biomedical domain as a case study, we specifically investigate and pre-train three knowledge adapters based on the UMLS Metathesaurus graph, the Wikipedia articles for diseases, and the semantic grouping information for biomedical concepts. Such adapters serve as off-the-shelf modules and can be used in a plug-and-play manner via DAKI.

- Extensive experiments on different biomedical NLP tasks and datasets demonstrate the benefits of the proposed knowledge-specific adapters and DAKI.

5.1.1 Related Work. This study is essentially related to two lines of research: knowledge integration for PLMs and domain-specific PLMs (biomedical PLMs in particular).

There has been a surge of research on knowledge injection for PLMs in recent years B. He, Jiang, Xiao, and Liu (2020); B. Kim et al. (2020); Lauscher et al. (2020); Levine et al. (2020); Pereira, Liu, Cheng, Asahara, and Kobayashi (2020); Peters et al. (2019); T. Sun et al. (2020a); X. Wang et al. (2021); Yao et al. (2019b); Z. Zhang et al. (2019). These studies aim to integrate knowledge from an

external knowledge source, e.g., Wikipedia, into PLMs by augmenting the training objective with a knowledge-driven regularization. As mentioned above, these methods are limited in the sense that they mostly focus on single-source knowledge, and require full model training. K-adapter R. Wang et al. (2020) addresses some of these issues by introducing linguistic and factual adapters into RoBERTa, but the adapters are treated equally in their work. Also, general domain knowledge, such as factual knowledge B. He et al. (2020); T. Sun et al. (2020a); X. Wang et al. (2021); Z. Zhang et al. (2019), commonsense knowledge Lauscher et al. (2020); Pereira et al. (2020), and linguistic knowledge Levine et al. (2020) are prioritized in these studies, while domain knowledge is somewhat underexplored Michalopoulos et al. (2020).

Biomedical NLP continues to be an active area of research in the past few years. There have been several biomedical PLMs proposed and have proven to be successful in various domain tasks Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019); Huang et al. (2019); Lee et al. (2020); Y. Peng et al. (2019a). As variants of BERT Devlin et al. (2019) in the biomedical domain, these PLMs are mostly pre-trained on large amounts of domain-specific corpora, such as the PubMed texts Lee et al. (2020); Y. Peng et al. (2019a) and clinical notes Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019); Huang et al. (2019), and do not explicitly incorporate domain knowledge in the pre-training stage.

This work differs from the aforementioned studies in that we are the first to integrate biomedical domain-specific knowledge from multiple sources into PLMs via an adapter-based architecture. The knowledge integration process is flexible, efficient, and transferable.

5.1.2 Diverse Adapters for Knowledge Integration (DAKI).

In this section, we introduce a mechanism, i.e., DAKI, that encodes domain knowledge from diverse sources into PLMs via knowledge-specific adapters. We first introduce the adapter module along with the overall architecture of DAKI, and then discuss the knowledge-specific adapters for the biomedical domain. In the end, we explain the attention-based knowledge controller that is leveraged to adaptively integrate these adapters.

5.1.2.1 Pre-trained Language Models with Adapters.

Adapter An *adapter* module is a simple and lightweight neural network placed within a large pre-trained base model, and in NLP the base model is usually a pre-trained language model such as BERT Devlin et al. (2019). Generally, adapters are placed in or between the intermediate transformer layers in a PLM, and the placement defines two paradigms. One puts the adapters *inside* the intermediate transformer layers Houlsby et al. (2019); Pfeiffer et al. (2021, 2020), and the other puts the adapter *between* and *outside* the intermediate transformer layers R. Wang et al. (2020). In this work, we choose the latter paradigm for its flexibility and extensibility, as shown in Figure 11. Instead of updating the entire language model, only the adapters are updated during fine-tuning on downstream tasks. This strategy demonstrates parameter-efficiency and scalability while achieving similar performance to full fine-tuning, and has been actively explored as an alternative for transfer learning in recent NLP studies Houlsby et al. (2019); Pfeiffer et al. (2021, 2020); Rücklé et al. (2020); R. Wang et al. (2020).

In this work, we leverage a simple yet effective bottleneck feed-forward network as the adapter module. Essentially, the adapter module consists of a residual connection and two projection layers with `LeakyReLU` as the activation,

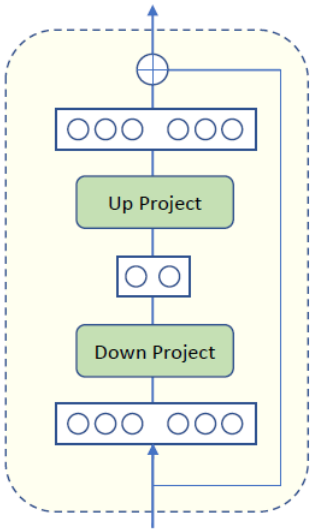


Figure 10. Adapter module.

as shown in Figure 10. The size of adapters is controlled by the bottleneck, and is usually much smaller than that of the base PLM, i.e., $d_{\text{bottleneck}} \ll d_{\text{PLM}}$, where d_{PLM} refers to the dimension of hidden-states in the base PLM. In our case, the bottleneck dimension is set to 128 for all experiments. Note that a more complex adapter is possible, such as two projection layers along with a stack of transformer layers R. Wang et al. (2020), but at the cost of efficiency.

Architecture Figure 11 illustrates the overall architecture of DAKI. Essentially, the architecture contains three main components, i.e., the base PLM, the knowledge-specific adapters, and the adapter integration module. DAKI theoretically supports any transformer-based structure as the base PLM, such as BERT Devlin et al. (2019), ALBERT Lan et al. (2019), RoBERTa Y. Liu et al. (2019), etc. Each knowledge-specific adapter contains several adapter modules and they are inserted at certain layers of the base PLM. Each adapter module takes as input the addition of the hidden-states of the transformer layer and the output of the previous adapter module. The adapter modules do not share weights with

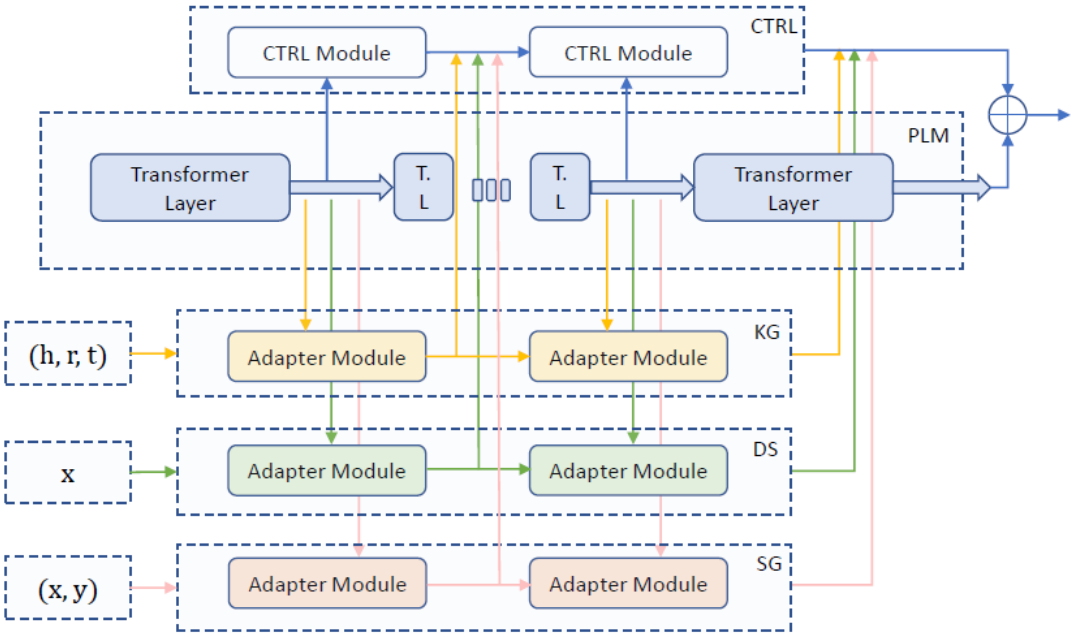


Figure 11. Architecture of DAKI. CTRL refers to the knowledge controller. Linear layers are omitted for simplicity.

each other. Motivated by the fact that knowledge from different sources should have different level of activation over downstream tasks, we incorporate a *knowledge controller* to adaptively integrate the knowledge adapters. Details are explained in Section 5.1.2.3.

When pre-training an adapter, we take the addition of the output of the last adapter module and the last-hidden-states of the base PLM as the final output, and use it for the pre-training task. Note that during adapter pre-training, the knowledge controller is dropped and the base PLM is frozen. When applying DAKI to downstream tasks, we take the addition of the output of the knowledge controller and the last-hidden-states of the base PLM as the final output, and use it for the downstream task.

The benefits of this architecture is threefold. First, adapters are independent and do not interact during pre-training, which means they have perfect memory of

Adapter	Source	Size	Format
KG	UMLS Metathesaurus	$1,772,248$	(h, r, t)
DS	Wikipedia	$14,617$	x
SG	Semantic Network	$333,005$	(x, y)

Table 24. Statistics of the datasets for pre-training KG, DS, SG. The formats are triples, passages, and textual definitions with labels, respectively.

the knowledge, thus avoiding the forgetting issue in multi-task learning. Second, it demonstrates flexibility and extensibility as it is easy to remove, add or replace the adapters. Third, the usage of DAKI is as simple as a general PLM, since its output can be considered the last-hidden-states of a PLM.

In this work, we use ALBERT-xxlarge-v2 Lan et al. (2019) as the base PLM. We investigate three knowledge-specific adapters based on the UMLS Metathesaurus graph, the Wikipedia articles for diseases, and the semantic grouping information for biomedical concepts. Details are explained in Section 5.1.2.2. Each adapter contains three adapter modules and they are placed at layers $\{0, 5, 11\}$. Note that the number and placement of adapter modules can be flexible, and in this study, we follow the same strategy with R. Wang et al. (2020) where three modules are distributed at the bottom, middle, and top layer.

5.1.2.2 Adapters Pre-training. In this work, we investigate three independent adapters based on three sources of knowledge, i.e., the UMLS Metathesaurus knowledge graph (KG), the Wikipedia articles for diseases (DS), and the semantic grouping information for medical concepts (SG). The statistics of the corresponding datasets for pre-training are shown in Table 24. These knowledge-specific adapters serve as examples for encoding domain knowledge from various sources, and can be easily extended or replaced with alternative knowledge

sources. For clarity, we use PLM-KG, PLM-DS and PLM-SG to denote the model that is used to pre-train the adapters in this section.

Knowledge Graph Adapter (KG) Knowledge graphs encode real-world knowledge in the form of triples, i.e., *(h, r, t)* where h and t refer to the head and tail entity and r is the relation between them. Knowledge graphs have been actively explored in recent studies of language model pre-training or fine-tuning, as they reveal the relationships between real-world entities that are hidden from surface texts.

To leverage the knowledge encoded in the UMLS Metathesaurus graph[1], we pre-train an adapter that aims to capture the connectivity patterns between medical entities through knowledge graph completion. More specifically, we treat the triples in UMLS as textual sequences and feed them into the PLM-KG encoder. Then the representation of the triple is used as input to a binary classification layer for plausibility prediction.

In particular, given a triple *(h, r, t)*, we first convert it to a textual sequence by concatenating the words in the names of h, r, and t. For example, for a triple *(diffuse adenocarcinoma of the stomach, disease has normal tissue origin, gastric mucosa)*, the constructed input sequence is:

`[CLS]` diffuse adenocarcinoma of the stomach `[SEP]` disease has normal tissue origin `[SEP]` gastric mucosa `[SEP]`

We then use the PLM-KG model to encode the sequence and use the representation for the `[CLS]` token in the last layer to predict the plausibility of the triple, i.e., determining whether the triple is valid or not. The adapter parameters

[1]The data is available at `https://www.nlm.nih.gov/research/umls`.

in this model are optimized with a binary cross-entropy loss:

$$\mathcal{L}_{\text{KG}} = -\sum_{t \in \{\mathcal{T}^+ \cup \mathcal{T}^-\}} (y \log \hat{y}_1 + (1-y) \log \hat{y}_0) \tag{5.1}$$

where y is the ground-truth label and $\hat{y}_0, \hat{y}_1$ refer to the output prediction probabilities. $\mathcal{T}^+$ and $\mathcal{T}^-$ are the positive and negative triple set. Here, the negative set $\mathcal{T}^-$ is constructed by replacing the head or tail entity in a positive triple with a random entity.

Disease Adapter (DS) It is crucial to equip pre-trained language models with disease knowledge for medical NLP tasks, as it bridges the gap between disease terms and their textual descriptions. For example, in the medical natural language inference task (NLI), the premise-hypothesis pair *(No history of blood clots or DVTs has never had chest pain prior to one week ago, Patient has angina)* is more likely to be correctly classified as `entailment` if the model specifically knows that angina refers to chest pain.

To leverage the disease knowledge, we pre-train an adapter that aims to infer disease names based on their textual descriptions. More specifically, for each disease, a new passage is formed by collecting the textual content from its Wikipedia article[2]. We then randomly substitute 75% of the *disease terms* in the passage with `[MASK]` in the passage and optimize the PLM-DS model via a masked language modeling (MLM) objective.

Formally, let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_K\}$ denote the indexes of the masked tokens in the passage T, where K is the number of masked tokens. Then T_{Π} and $T_{\neg\Pi}$ represent the set of masked and observed tokens in the passage, respectively. Then

[2]This data is proposed by (Y. He et al., 2020a).

the training objective for the adapter parameters is described as:

$$\mathcal{L}_{\text{DS}} = \mathcal{L}_{\text{mlm}}(T_{\Pi}|T_{\neg\Pi}) = -\frac{1}{K}\sum_{k=1}^{K}\log p(t_{\pi_k}|T_{\neg\Pi}) \tag{5.2}$$

where $p(t_{\pi_k}|T_{\neg\Pi})$ is the probability of predicting t_{π_k} given the unmasked tokens $T_{\neg\Pi}$, estimated by a softmax layer.

Semantic Grouping Adapter (SG) To provide a proper and consistent categorization of concepts in the Metathesaurus, the UMLS Semantic Network groups concepts according to the semantic types that have been assigned to them. Each concept is assigned to at least one semantic type from a total of 127 semantic types. For certain purposes, however, a coarser-grained categorization is desirable, and hence the semantic types are aggregated into 15 semantic groupings McCray, Burgun, and Bodenreider (2001). Such aggregation ensures the semantic coherence between concepts in the same group[3]. This property would help pre-trained language models capture the connectivity between medical concepts, as well as between their descriptive texts.

To leverage the semantic grouping information, we pre-train an adapter that aims to predict the semantic groupings of concepts in UMLS based on their textual definitions. More specifically, for a UMLS concept with a corresponding textual definition, we encode the definition with the PLM-SG model and feed the `[CLS]` representation into a linear layer for classification. The model is optimized with cross-entropy loss:

$$\mathcal{L}_{\text{SG}} = -\sum_{i=1}^{15} y_i \log \hat{y}_i \tag{5.3}$$

where y_i is the ground-truth label and $\hat{y}_i$ refers to the output prediction probabilities.

[3]The data is available at `https://semanticnetwork.nlm.nih.gov`.

5.1.2.3 *Knowledge Controller.*

The *knowledge controller* is essentially a separate adapter with additional linear layers, which is distributed at the same layers with the knowledge adapters, as shown in Figure 11. This module aims to adaptively integrate the knowledge adapters by assigning them different importance weights, as opposed to a simple concatenation of the outputs of adapters R. Wang et al. (2020). At each layer i where an adapter module is placed, three linear transformation modules are employed, i.e., Q_i, K_i, V_i, as motivated by Vaswani et al. (2017). Essentially, Q_i takes the hidden-states of the controller as the input, and the output is considered as the *query* signal. K_i in contrast takes the hidden-states of the adapters as the input, and the output serves as the *key* signal. The *value* signal is the hidden-states of the adapters. Then the attention weights are computed for each adapter and the weighted sum of the hidden-states of adapters are fed into V_i, the output of which is regarded as the final output of the knowledge controller at layer i:

$$\begin{aligned} \mathbf{Q}_i &= \mathbf{W}_{Q_i}\mathbf{H}_{C_i} + \mathbf{b}_{Q_i} \\ \mathbf{K}_i &= \mathbf{W}_{K_i}\mathbf{H}_{D_i} + \mathbf{b}_{K_i} \\ \mathbf{A}_i &= \text{softmax}(\mathbf{Q}_i\mathbf{K}_i^T)\mathbf{H}_{D_i} \\ \mathbf{Z}_i &= \mathbf{W}_{V_i}\mathbf{A}_i + \mathbf{b}_{V_i} \end{aligned} \tag{5.4}$$

where $\mathbf{H}_{C_i}$ are the hidden-states of the controller and $\mathbf{H}_{D_i}$ are the concatenation of the hidden-states of the adapters at layer i. $\mathbf{W}_{Q_i}, \mathbf{b}_{Q_i}, \mathbf{W}_{K_i}, \mathbf{b}_{K_i}, \mathbf{W}_{V_i}, \mathbf{b}_{V_i}$ are trainable parameters of the linear modules at each layer.

5.1.3 Experiments.

In this section, we evaluate the DAKI architecture over three knowledge-driven downstream tasks in biomedical NLP, where we aim to show the effectiveness of the knowledge integration method. We also investigate some desirable properties of the architecture.

5.1.3.1 Setup. We perform evaluation over three knowledge-driven biomedical NLP tasks, i.e., Question Answering (QA), Natural Language Inference (NLI) and Named Entity Recognition (NER)[4].

QA We conduct the medical QA experiments on MEDIQA-2019 Abacha, Shivade, and Demner-Fushman (2019) and TRECQA-2017 Abacha, Agichtein, Pinter, and Demner-Fushman (2017), where the task is cast as a regression problem. Essentially, for a given question-answer pair, a numerical score ranging from -2 to 2 is assigned by experts, indicating the quality of the answer to the question, and the task is to predict the score. We use a simple prediction model, where each pair is encoded with a PLM or DAKI, and the representation for `[CLS]` is fed into a linear layer on top for prediction.

NLI We conduct the medical NLI experiments on MEDNLI Romanov and Shivade (2018a), where the task is to classify a given premise-hypothesis pair into a class of `entailment`, `neutral`, or `contradiction`. Similarly, each pair is encoded with a PLM or DAKI, and the `[CLS]` representation is fed into a classification head on top.

NER We conduct the medical NER experiments on NCBI Doğan et al. (2014) and BC5CDR-disease Wei et al. (2016), where the task is to classify tokens of sentences into a class of `B`, `I`, or `O` Y. He et al. (2020a); Y. Peng et al. (2019a), with a PLM or DAKI as the encoder.

Note that our models for downstream tasks QA, NLI, and NER follow those in diseaseBERT/diseaseALBERT Y. He et al. (2020a) to be comparable. We

[4]The datasets for downstream tasks are available at `https://github.com/heyunh2015/diseaseBERT`.

also inherit the hyper-parameters for such models from Y. He et al. (2020a). In particular, we employ AdamW as the optimizer and set learning rates of {1e-5, 1e-5, 5e-5}, and the batch sizes of {8, 16, 16} respectively for the tasks.

Baselines We take three PLMs, i.e., BERT-base-uncased Devlin et al. (2019), RoBERTa-base Y. Liu et al. (2019), ALBERT-xxlarge-v2 Lan et al. (2019), as well as their main biomedical variants as the baselines, including ClinicalBERT Alsentzer, Murphy, Boag, Weng, Jindi, et al. (2019), SciBERT Beltagy et al. (2019), BioBERT-v1.1 Lee et al. (2020), umlsBERT Michalopoulos et al. (2020) and diseaseBERT/diseaseALBERT Y. He et al. (2020a).

Pre-training Adapters When pre-training the adapters KG, DS, SG, we use the ALBERT-xxlarge-v2 Lan et al. (2019) as the base PLM, and set the adapter size to 128. We use Adam as the optimizer and set learning rates of {1e-6, 2e-4, 1e-5}, batch sizes of {256, 16, 256}, maximum sequence lengths of {16, 256, 128} and training epochs of {2, 10, 1}, respectively for the corresponding adapters.

5.1.3.2 Results. Table 25 shows the performance of our proposed architecture, i.e., DAKI, over three biomedical NLP tasks across five datasets. Generally, one main observation from the table is that equipping PLMs with DAKI significantly improve their performance on these biomedical tasks, as reflected in DAKI-BERT, DAKI-RoBERTa and DAKI-ALBERT, demonstrating the effectiveness of the architecture. Moreover, although DAKI-BERT outperforms BERT across all metrics, it only performs comparably or poorer than ClinicalBERT, SciBERT and BioBERT. We conjecture that it is due to lack of the knowledge in their pre-training data, i.e., the MIMIC-III clinical notes Johnson

Datasets	MEDIQA-2019			TRECQA-2017			MEDNLI	BC5CDR	NCBI
Metrics(%)	Accuracy	MRR	Precision	Accuracy	MRR	Precision	Accuracy	F1	F1
BERT	64.95	82.72	66.49	74.61	56.17	52.55	75.95	83.09	85.14
ClinicalBERT	67.30	84.78	70.59	77.00	52.56	56.62	81.50	84.90	87.25
SciBERT	68.47	84.47	68.07	77.23	54.57	57.54	80.94	86.16	87.24
BioBERT	68.29	83.61	72.78	77.12	49.84	57.25	81.86	85.99	87.70
diseaseBERT	66.40	83.33	68.94	75.33	56.41	54.01	77.29	83.47	86.81
umlsBERT	62.87	83.91	63.62	70.20	54.17	46.69	81.65	84.54	86.23
RoBERTa	72.49	86.74	74.67	75.33	51.76	54.01	81.65	83.04	85.83
ALBERT	76.54	88.46	81.41	75.09	58.57	53.03	85.48	84.28	87.56
diseaseALBERT	79.49	90.00	84.02	80.10	57.21	**62.40**	86.15	84.71	87.69
DAKI-BERT	69.47	85.06	70.17	77.95	54.65	58.27	77.85	83.43	85.67
DAKI-BioBERT	72.54	87.33	77.46	78.55	54.17	59.04	83.41	**86.51**	**89.01**
DAKI-RoBERTa	73.98	89.22	76.39	77.23	51.92	58.48	81.65	83.36	86.01
DAKI-ALBERT	**80.22**	**91.22**	**84.36**	**80.33**	**58.65**	62.31	**86.85**	84.86	87.86

Table 25. Performance of DAKI over downstream tasks QA, NLI and NER.

et al. (2016), the Semantic Scholar papers Ammar et al. (2018), and the PubMed articles, respectively.

Transferability Another advantage of DAKI is transferability, due to its flexible architecture and implementation. In this work, we have three adapters and they are all pre-trained with ALBERT as the base PLM. All the DAKI variants in Table 25 are the corresponding PLMs equipped with such pre-trained adapters (based on ALBERT). As such, the performance gain of the DAKI variants shows that the knowledge in the adapters is transferable across BERT versions, making it possible to use adapters as off-the-shelf modules in a plug-and-play manner. Interestingly, even for the knowledge-augmented BioBERT, incorporating DAKI yields a performance boost over all tasks, which further demonstrates the transferability of the architecture.

Ablation Study To investigate the influence of each component of DAKI, we perform an ablation study and show the results in Table 26. We first remove the

Datasets	MEDIQA-2019			TRECQA-2017			MEDNLI	BC5CDR	NCBI	A.P	C.P
Metrics(%)	Acc	MRR	Pre	Acc	MRR	Pre	Acc	F1	F1		
DAKI	80.22	91.22	84.36	80.33	58.65	62.31	86.85	84.86	87.86	79.63	-
w/o ctrl	78.32	88.27	81.68	79.38	56.09	61.19	86.78	84.58	86.99	78.14	-1.49
w/o KG	79.49	90.72	85.42	80.45	57.85	62.74	85.94	83.93	87.43	79.33	-0.30
w/o DS	78.86	89.61	82.37	79.62	57.85	61.53	85.86	83.99	87.82	78.61	-1.02
w/o SG	73.15	86.33	80.77	79.26	57.61	60.43	85.37	84.29	86.87	77.12	-2.51
w/o ctrl,DS,SG	78.14	89.61	80.11	79.86	59.13	62.11	86.29	83.76	87.37	78.48	-1.15
w/o ctrl,KG,SG	77.78	89.44	83.54	79.98	57.45	61.96	84.18	83.46	87.33	78.34	-1.29
w/o ctrl,KG,DS	77.51	89.83	83.44	80.69	58.01	64.01	86.51	84.25	87.26	79.05	-0.58
ALBERT	76.15	84.67	83.19	77.12	57.93	56.68	86.01	85.38	86.81	77.10	-2.53

A.P means average of performance and C.P means change of performance.

Table 26. Ablation analysis.

knowledge controller from DAKI, and take the addition of the outputs of adapters, without adaptive adjustment. Then we remove each adapter while keeping the controller. Finally, we apply accumulative ablation by removing both of them. Essentially, the results of the ablated versions demonstrate varying degrees of performance drop, indicating the necessity of each component.

Explainability We expect the *knowledge controller* to bring some explainability, as it adaptively activates the adapters when fine-tuning over the downstream tasks. We show the average softmax attention weights of the adapters in Figure 12, which we assume to reflect the activation levels of them. Basically, the activations of adapters are different across tasks and datasets, except that KG and SG seem to have more impact on BC5CDR and NCBI.

Parameter-efficiency An advantage of using DAKI for incorporating knowledge is that only one version of the PLM is needed to accommodate multiple knowledge sources. In particular, without adapters, fine-tuning a PLM with one knowledge source will produce a new version of PLM. For three knowledge sources in our work, we will need to have $3 \times N_{\text{PLM}}$ parameters. With DAKI, this number

Figure 12. Activation levels of the adapters KG, DS, SG over the downstream tasks. We calculate the softmax activations in the last layer for each adapter, and the activations are averaged over all instances in the test set.

is reduced to $N_{\text{PLM}} + 3 \times N_{\text{adapter}} + N_{\text{ctrl}}$. Considering ALBERT as an example, this amount to a reduction of $2 \times N_{\text{PLM}} - 3 \times N_{\text{adapter}} - N_{\text{ctrl}} \approx 2 \times 223M - 4M = 442M$ parameters.

5.2 Conclusion

In this section, we propose DAKI, an adapter-based architecture that adaptively integrates knowledge from multiple sources into pre-trained language models. We take the biomedical domain as a case study, and specifically explore three different sources of biomedical knowledge and integrate them with DAKI. The experimental results prove the effectiveness of the architecture and also show that the architecture demonstrates parameter-efficiency, transferability, and explainability to some degree. The objective of this work is not to update state-of-the-art results on the benchmarks but to provide an alternative method of domain knowledge integration for PLMs, especially from multiple sources of knowledge.

CHAPTER VI

CONCLUSION AND FUTURE DIRECTIONS

6.1 Conclusion

In conclusion, this dissertation has investigated a range of approaches to enhance pre-trained language models in the clinical domain, with a specific focus on knowledge graphs, data augmentation, and parameter-efficient domain knowledge integration. These explorations have shed light on the potential of leveraging these techniques to improve the performance and applicability of language models in healthcare settings. The focus of this dissertation is to demonstrate effective techniques for incorporating domain knowledge in healthcare. Each chapter represents an exploration of different approaches and innovations, each contributing to the body of knowledge in clinical NLP and healthcare informatics. This final chapter offers an overview of the major contributions and findings, along with a discussion of potential directions for future research.

In Chapter II, we perform a comprehensive review of existing clinical PLMs, scrutinizing their performances and providing a critical analysis of potential improvement areas. This review serves as a foundation for the subsequent exploration of enhancement strategies, setting the stage for the experimental chapters that followed.

In Chapter III, we delve into the application of graph representation learning techniques to integrate internal and external knowledge graphs into healthcare machine learning models. We propose three unique studies: the use of hyperbolic embeddings of medical ontologies, a network embedding method for maintaining network view consistency, and a method for extracting medical text

from EHRs for patient outcome prediction. The potential of these methods to enhance PLMs is thoroughly demonstrated.

In Chapter IV, we explore the potential of clinical text as a source of domain knowledge, proposing two innovative methods of data augmentation. We introduce a framework for generating synthetic clinical notes, MedAug, which demonstrates significant potential to enhance patient outcome prediction models. Furthermore, we propose ClinicalT5, a domain-specific T5-based transformer model pre-trained on clinical text, which demonstrates superior performance in various clinical tasks.

In Chapter V, we present a novel, parameter-efficient approach to incorporating domain knowledge of multiple sources and formats into PLMs using adapters. These small, feed-forward networks encode domain-specific knowledge and improve the performance of various biomedical NLP tasks. Importantly, this approach allows the language models to retain their original general-domain knowledge, ensuring their robustness and adaptability.

6.2 Future Directions

Given the contributions and findings of this dissertation, the journey into improving the performance of PLMs within the clinical domain is far from complete. Several potential avenues of research can be explored as future directions.

Large Language Models The advent of large language models (LLMs) like GPT-4 has significantly transformed the landscape of natural language processing. These models are equipped to discern nuanced patterns and generate highly context-specific responses. However, there's a recognized need for a more specialized approach in the clinical domain. General-domain LLMs, despite their broad contextual understanding, tend to underperform in comparison with small,

specialized clinical models in domain-specific tasks Lehman et al. (2023) and lack the depth of scientific and medical knowledge needed for the intricacies of disease mechanisms and treatments Ruksakulpiwat, Kumar, and Ajibade (2023). This observed limitation paves the way for exciting research opportunities focused on the development of domain-specific clinical LLMs. Although there have been initial endeavors like Med-PaLM Singhal et al. (2023), the field is far from mature.

The sensitive and private nature of medical data has resulted in a scarcity of data for training clinical LLMs. To address this, potential approaches could include the fine-tuning of general LLMs with doctor-patient conversations Yunxiang, Zihan, Kai, Ruilong, and You (2023), thereby creating domain-specific variants with limited data. Data augmentation techniques, like the generation of synthetic data using GPT-4, can also be explored. Nonetheless, these approaches are not without challenges. One such challenge is hallucination where the model generates incorrect or misleading information. This could have severe implications in clinical settings, emphasizing the need for high-quality data and the incorporation of additional information sources, such as knowledge graphs, to improve model accuracy and explainability.

Knowledge Graphs Knowledge Graphs (KGs) represent another promising direction in the field of clinical NLP and health informatics. As structured representations of interconnected data, KGs provide a powerful means of organizing, interpreting, and employing domain-specific knowledge. This ability is particularly pertinent in the medical field, where intricate relationships exist between entities like diseases, symptoms, medications, and genetic factors. However, despite their potential, current applications of KGs in the healthcare sector remain relatively underexplored.

The main challenge lies in the construction and maintenance of large-scale, high-quality, and up-to-date KGs that can encompass rapidly evolving medical knowledge. Current solutions mostly rely on manual curation which is time-consuming, expensive, and struggle to keep up with the latest research findings and clinical guidelines. To overcome this issue, one possible solution is to develop automated methods that can extract relevant knowledge from various sources, including clinical literature, Electronic Health Records (EHRs), and other databases. This could facilitate the construction of robust, data-driven KGs. Moreover, while the existing work on KGs in clinical NLP focuses largely on their use for enhancing model performance, another potential direction is to improve the interpretability and explainability of complex models with KGs, which is crucial in the clinical domain.

Multimodality Multimodality presents another fascinating future direction in this field, which in this context refers to the incorporation and analysis of diverse types of data - including text, images, numerical lab results, and more. This approach aligns well with the heterogeneous nature of healthcare data. Electronic Health Records (EHRs), for example, are a rich source of multimodal data. They contain a wide range of information, including physician's notes, medical imaging, lab results and patient demographics, all of which could provide valuable insights when leveraged together in the era of large language models. Two recent studies, Med-PaLM M Tu et al. (2023) and BiomedGPT K. Zhang et al. (2023), have indeed shown the potential of integrating multimodal data in healthcare informatics. However, this direction is far from being comprehensively explored and thus, presents numerous exciting opportunities for future research.

REFERENCES CITED

Abacha, A. B., Agichtein, E., Pinter, Y., & Demner-Fushman, D. (2017). Overview of the medical question answering task at trec 2017 liveqa. In *Proceedings of the text retrieval conference (trec).*

Abacha, A. B., Shivade, C., & Demner-Fushman, D. (2019). Overview of the mediqa 2019 shared task on textual inference, question entailment and question answering. In *Proceedings of the 18th bionlp workshop and shared task (bionlp)* (pp. 370–379).

Allada, A. K., Wang, Y., Jindal, V., Babee, M., Tizhoosh, H. R., & Crowley, M. (2021). Analysis of language embeddings for classification of unstructured pathology reports. In *2021 43rd annual international conference of the ieee engineering in medicine & biology society (embc)* (pp. 2378–2381).

Alsentzer, E., Murphy, J., Boag, W., Weng, W.-H., Jin, D., Naumann, T., & McDermott, M. (2019, June). Publicly available clinical BERT embeddings. In *Proceedings of the 2nd clinical natural language processing workshop* (pp. 72–78). Minneapolis, Minnesota, USA: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/W19-1909` doi: 10.18653/v1/W19-1909

Alsentzer, E., Murphy, J., Boag, W., Weng, W.-H., Jindi, D., Naumann, T., & McDermott, M. (2019). Publicly available clinical bert embeddings. In *Proceedings of the 2nd clinical natural language processing workshop (clinicalnlp)* (pp. 72–78).

Amin-Nejad, A., Ive, J., & Velupillai, S. (2020, May). Exploring transformer text generation for medical dataset augmentation. In *Proceedings of the 12th language resources and evaluation conference* (pp. 4699–4708). Marseille, France: European Language Resources Association. Retrieved from `https://aclanthology.org/2020.lrec-1.578`

Ammar, W., Groeneveld, D., Bhagavatula, C., Beltagy, I., Crawford, M., Downey, D., ... others (2018). Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 3, industry papers (naacl-hlt)* (pp. 84–91).

Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., ... Zwerdling, N. (2020, Apr.). Do not have enough data? deep learning to the rescue! *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 7383-7390. Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/6233 doi: 10.1609/aaai.v34i05.6233

Aronson, A. R., & Lang, F.-M. (2010). An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, *17*(3), 229–236.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Baechle, C., Agarwal, A., Behara, R., & Zhu, X. (2017). Latent topic ensemble learning for hospital readmission cost reduction. In *2017 international joint conference on neural networks (ijcnn)* (pp. 4594–4601).

Baker, S., Silins, I., Guo, Y., Ali, I., Högberg, J., Stenius, U., & Korhonen, A. (2016). Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics*, *32*(3), 432–440.

Basaldella, M., Liu, F., Shareghi, E., & Collier, N. (2020, November). COMETA: A corpus for medical entity linking in the social media. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 3122–3137). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.emnlp-main.253 doi: 10.18653/v1/2020.emnlp-main.253

Beltagy, I., Lo, K., & Cohan, A. (2019). Scibert: A pretrained language model for scientific text. 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)* (pp. 3606–3611).

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Bengio, Y., Ducharme, R., & Vincent, P. (2000). A neural probabilistic language model. *Advances in neural information processing systems*, *13*.

Bhowmik, R., Stratos, K., & de Melo, G. (2021, April). Fast and effective biomedical entity linking using a dual encoder. In *Proceedings of the 12th international workshop on health text mining and information analysis* (pp. 28–37). online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.louhi-1.4

Bodenreider, O. (2004). The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, *32*(suppl_1), D267–D270.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. Retrieved from https://aclanthology.org/Q17-1010 doi: 10.1162/tacl_a_00051

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems* (pp. 2787–2795).

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... others (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Cao, S., Lu, W., & Xu, Q. (2015). Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th acm international on conference on information and knowledge management* (pp. 891–900).

Cao, S., Lu, W., & Xu, Q. (2016). Deep neural networks for learning graph representations. In *Thirtieth aaai conference on artificial intelligence.*

Chakraborty, S., Bisong, E., Bhatt, S., Wagner, T., Elliott, R., & Mosconi, F. (2020). Biomedbert: A pre-trained biomedical language model for qa and ir. In *Proceedings of the 28th international conference on computational linguistics* (pp. 669–679).

Chang, J., & Blei, D. (2009). Relational topic models for document networks. In *Artificial intelligence and statistics* (pp. 81–88).

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Chen, H., Hong, P., Han, W., Majumder, N., & Poria, S. (2020). Dialogue relation extraction with document-level heterogeneous graph attention networks. *arXiv preprint arXiv:2009.05092*.

Chen, J., & Yang, D. (2021, June). Structure-aware abstractive conversation summarization via discourse and action graphs. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 1380–1391). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.naacl-main.109 doi: 10.18653/v1/2021.naacl-main.109

Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns. *Transactions of the association for computational linguistics*, *4*, 357–370.

Choi, Y., Chiu, C. Y.-I., & Sontag, D. (2016). Learning low-dimensional representations of medical concepts. *AMIA Summits on Translational Science Proceedings*, *2016*, 41.

Christopoulou, F., Miwa, M., & Ananiadou, S. (2019). Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. 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)* (pp. 4927–4938). Association for Computational Linguistics.

Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Iclr.* Retrieved from `https://openreview.net/pdf?id=r1xMH1BtvB`

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... Stoyanov, V. (2020, July). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 8440–8451). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.acl-main.747` doi: 10.18653/v1/2020.acl-main.747

Cui, Y., Che, W., Liu, T., Qin, B., & Yang, Z. (2021). Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3504–3514.

Dai, Q., Li, Q., Tang, J., & Wang, D. (2018). Adversarial network embedding. In *Thirty-second aaai conference on artificial intelligence.*

Dai, Z., Lai, G., Yang, Y., & Le, Q. (2020). Funnel-transformer: Filtering out sequential redundancy for efficient language processing. *Advances in neural information processing systems*, *33*, 4271–4282.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., & Salakhutdinov, R. (2019, July). Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 2978–2988). Florence, Italy: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/P19-1285` doi: 10.18653/v1/P19-1285

Davison, B. A., Metra, M., Senger, S., Edwards, C., Milo, O., Bloomfield, D. M., ... others (2016). Patient journey after admission for acute heart failure: length of stay, 30-day readmission and 90-day mortality. *European journal of heart failure*, *18*(8), 1041–1050.

De Cao, N., Izacard, G., Riedel, S., & Petroni, F. (2021). Autoregressive entity retrieval. In *International conference on learning representations.* Retrieved from `https://openreview.net/forum?id=5k8F6UU39V`

De Cao, N., Aziz, W., & Titov, I. (2019). Question answering by reasoning across documents with graph convolutional networks. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 2306–2317).

De Cao, N., Wu, L., Popat, K., Artetxe, M., Goyal, N., Plekhanov, M., ... Petroni, F. (2022). Multilingual autoregressive entity linking. *Transactions of the Association for Computational Linguistics*, *10*, 274–290. Retrieved from `https://aclanthology.org/2022.tacl-1.16` doi: 10.1162/tacl_a_00460

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (naacl-hlt)* (pp. 4171–4186).

Dhingra, B., Shallue, C. J., Norouzi, M., Dai, A. M., & Dahl, G. E. (2018). Embedding text in hyperbolic spaces. *NAACL HLT 2018*, 59.

Ding, M., Zhou, C., Yang, H., & Tang, J. (2020). Cogltx: Applying bert to long texts. *Advances in Neural Information Processing Systems*, *33*.

Doğan, R. I., Leaman, R., & Lu, Z. (2014). Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, *47*, 1–10.

Du, N., Huang, Y., Dai, A. M., Tong, S., Lepikhin, D., Xu, Y., ... others (2022). Glam: Efficient scaling of language models with mixture-of-experts. In *International conference on machine learning* (pp. 5547–5569).

Fedus, W., Zoph, B., & Shazeer, N. (2021). *Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity.*

Feng, J., Phillips, R. V., Malenica, I., Bishara, A., Hubbard, A. E., Celi, L. A., & Pirracchio, R. (2022). Clinical artificial intelligence quality improvement: towards continual monitoring and updating of ai algorithms in healthcare. *npj Digital Medicine*, *5*(1), 1–9.

Gao, H., & Huang, H. (2018). Deep attributed network embedding. In *Ijcai* (Vol. 18, pp. 3364–3370).

Gao, Y., Dligach, D., Christensen, L., Tesch, S., Laffin, R., Xu, D., ... Afshar, M. (2022). A scoping review of publicly available language tasks in clinical natural language processing. *Journal of the American Medical Informatics Association*, *29*(10), 1797–1806.

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *International conference on machine learning* (pp. 1243–1252).

Geigle, C., Mei, Q., & Zhai, C. (2018). Feature engineering for text data. In *Feature engineering for machine learning and data analytics* (pp. 15–54). CRC Press.

Gonzalez-Hernandez, G., Sarker, A., O'Connor, K., & Savova, G. (2017). Capturing the patient's perspective: a review of advances in natural language processing of health-related text. *Yearbook of medical informatics*, *26*(01), 214–227.

Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 855–864).

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., ... Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, *3*(1), 1–23.

Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., ... Wu, Y. (2023). How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., & Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of acl*.

Hackbarth, G. (2009). Reforming america's health care delivery system. *Statement before the Senate Finance Committee Roundtable on Reforming America's Health Care Delivery System*, 5.

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in neural information processing systems* (pp. 1024–1034).

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*.

Hao, B., Zhu, H., & Paschalidis, I. C. (2020). Enhancing clinical bert embedding using a biomedical knowledge base. In *28th international conference on computational linguistics (coling 2020)*.

Harutyunyan, H., Khachatrian, H., Kale, D. C., & Galstyan, A. (2017). Multitask learning and benchmarking with clinical time series data. *CoRR*, *abs/1703.07771*.

He, B., Jiang, X., Xiao, J., & Liu, Q. (2020). Kgplm: Knowledge-guided language model pre-training via generative and discriminative learning. *arXiv preprint arXiv:2012.03551*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

He, Y., Zhu, Z., Zhang, Y., Chen, Q., & Caverlee, J. (2020a). Infusing disease knowledge into bert for health question answering, medical inference and disease name recognition. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 4604–4614).

He, Y., Zhu, Z., Zhang, Y., Chen, Q., & Caverlee, J. (2020b, November). Infusing Disease Knowledge into BERT for Health Question Answering, Medical Inference and Disease Name Recognition. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 4604–4614). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.emnlp-main.372` doi: 10.18653/v1/2020.emnlp-main.372

Henry, J., Pylypchuk, Y., Searcy, T., Patel, V., et al. (2016). Adoption of electronic health record systems among us non-federal acute care hospitals: 2008–2015. *ONC data brief*, *35*(35), 2008–2015.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. In *Nips deep learning and representation learning workshop.* Retrieved from `http://arxiv.org/abs/1503.02531`

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

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... Gelly, S. (2019). Parameter-efficient transfer learning for nlp. In *Proceedings of the international conference on machine learning (icml)* (pp. 2790–2799).

Huang, K., Altosaar, J., & Ranganath, R. (2019). Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

Huang, K., Singh, A., Chen, S., Moseley, E., Deng, C.-Y., George, N., & Lindvall, C. (2020, November). Clinical XLNet: Modeling sequential clinical notes and predicting prolonged mechanical ventilation. In *Proceedings of the 3rd clinical natural language processing workshop* (pp. 94–100). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.clinicalnlp-1.11` doi: 10.18653/v1/2020.clinicalnlp-1.11

Jiang, H., Gurajada, S., Lu, Q., Neelam, S., Popa, L., Sen, P., ... Gray, A. (2021, August). LNN-EL: A neuro-symbolic approach to short-text entity linking. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)* (pp. 775–787). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.acl-long.64` doi: 10.18653/v1/2021.acl-long.64

Jiang, Z.-H., Yu, W., Zhou, D., Chen, Y., Feng, J., & Yan, S. (2020). Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, *33*, 12837–12848.

Jin, Q., Dhingra, B., Cohen, W., & Lu, X. (2019). Probing biomedical embeddings from language models. In *Proceedings of the 3rd workshop on evaluating vector space representations for nlp (repeval)* (pp. 82–89).

Johnson, A. E., Kramer, A. A., & Clifford, G. D. (2014). Data preprocessing and mortality prediction: the Physionet/CinC 2012 challenge revisited. In *Computing in cardiology conference (cinc)* (p. 157-160).

Johnson, A. E., Pollard, T. J., Shen, L., Li-Wei, H. L., Feng, M., Ghassemi, M., ... Mark, R. G. (2016). Mimic-iii, a freely accessible critical care database. *Scientific data*, *3*(1), 1–9.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., & Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, *8*, 64–77.

JR, L. G., P, L., A, A., P, G., C, G., D, M., ... D, V. (1984). A simplified acute physiology score for icu patients. *Crit Care Med*, *12*(11), 975-977.

JR, L. G., S, L., & F, S. (1993). A new simplified acute physiology score (SAPS II) based on a european/north american multicenter study. *JAMA*, *270*(24), 2957-2963.

Junqueira, A. R. B., Mirza, F., & Baig, M. M. (2019). A machine learning model for predicting icu readmissions and key risk factors: analysis from a longitudinal health records. *Health and Technology*, *9*(3), 297–309.

Kalyan, K. S., Rajasekharan, A., & Sangeetha, S. (2021). Ammu: a survey of transformer-based biomedical pretrained language models. *Journal of biomedical informatics*, 103982.

Kalyan, K. S., & Sangeetha, S. (2020). Secnlp: A survey of embeddings in clinical natural language processing. *Journal of biomedical informatics*, *101*, 103323.

Kanakarajan, K. r., Kundumani, B., & Sankarasubbu, M. (2021, June). BioELECTRA:pretrained biomedical text encoder using discriminators. In *Proceedings of the 20th workshop on biomedical language processing* (pp. 143–154). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.bionlp-1.16` doi: 10.18653/v1/2021.bionlp-1.16

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., & Socher, R. (2019). Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Kim, B., Hong, T., Ko, Y., & Seo, J. (2020). Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th international conference on computational linguistics (coling)* (pp. 1737–1743).

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1746–1751).

Kipf, T. N., & Welling, M. (2016a). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kipf, T. N., & Welling, M. (2016b). Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Kitaev, N., Kaiser, Ł., & Levskaya, A. (2020). Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Kraljevic, Z., Shek, A., Bean, D., Bendayan, R., Teo, J., & Dobson, R. (2021). Medgpt: Medical concept prediction from clinical narratives. *arXiv preprint arXiv:2107.03134*.

Krompaß, D., Esteban, C., Tresp, V., Sedlmayr, M., & Ganslandt, T. (2015). Exploiting latent embeddings of nominal clinical data for predicting hospital readmission. *KI-Künstliche Intelligenz*, *29*(2), 153–159.

Kudo, T. (2018, July). Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 66–75). Melbourne, Australia: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/P18-1007` doi: 10.18653/v1/P18-1007

Kudo, T., & Richardson, J. (2018, November). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 conference on empirical methods in natural language processing: System demonstrations* (pp. 66–71). Brussels, Belgium: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/D18-2012` doi: 10.18653/v1/D18-2012

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent convolutional neural networks for text classification. In *Proceedings of the twenty-ninth aaai conference on artificial intelligence* (pp. 2267–2273).

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. In *Proceedings of the international conference on learning representations (iclr).*

Lange, L., Adel, H., Strötgen, J., & Klakow, D. (2022). Clin-x: pre-trained language models and a study on cross-task transfer for concept extraction in the clinical domain. *Bioinformatics*, *38*(12), 3267–3274.

Lauscher, A., Majewska, O., Ribeiro, L. F., Gurevych, I., Rozanov, N., & Glavaš, G. (2020). Common sense or world knowledge? investigating adapter-based knowledge injection into pretrained transformers. In *Proceedings of deep learning inside out (deelio): The first workshop on knowledge extraction and integration for deep learning architectures* (pp. 43–49).

Leacock, C., & Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, *49*(2), 265–283.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

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

Leeuwenberg, A., & Moens, M. F. (2018). Temporal information extraction by predicting relative time-lines. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 1237–1246).

Lehman, E., Hernandez, E., Mahajan, D., Wulff, J., Smith, M. J., Ziegler, Z., ... Alsentzer, E. (2023). Do we still need clinical language models? *arXiv preprint arXiv:2302.08091*.

Leimeister, M., & Wilson, B. J. (2018). Skip-gram word embeddings in hyperbolic space. *arXiv preprint arXiv:1809.01498*.

Levine, Y., Lenz, B., Dagan, O., Ram, O., Padnos, D., Sharir, O., ... Shoham, Y. (2020). Sensebert: Driving some sense into bert. In *Proceedings of the 58th annual meeting of the association for computational linguistics (acl)* (pp. 4656–4667).

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... Zettlemoyer, L. (2020). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7871–7880).

Lewis, P., Ott, M., Du, J., & Stoyanov, V. (2020). Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art. In *Proceedings of the 3rd clinical natural language processing workshop* (pp. 146–157).

Li, F., Jin, Y., Liu, W., Rawat, B. P. S., Cai, P., Yu, H., et al. (2019). Fine-tuning bidirectional encoder representations from transformers (bert)–based models on large-scale electronic health record notes: an empirical study. *JMIR medical informatics*, *7*(3), e14830.

Li, J., Sun, Y., Johnson, R. J., Sciaky, D., Wei, C.-H., Leaman, R., ... Lu, Z. (2016). Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, *2016*.

Li, Y., Rao, S., Solares, J. R. A., Hassaine, A., Ramakrishnan, R., Canoy, D., ... Salimi-Khorshidi, G. (2020). Behrt: transformer for electronic health records. *Scientific reports*, *10*(1), 1–12.

Li, Y., Tarlow, D., Brockschmidt, M., & Zemel, R. (2015). Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*.

Li, Y., Wehbe, R. M., Ahmad, F. S., Wang, H., & Luo, Y. (2022). Clinical-longformer and clinical-bigbird: Transformers for long clinical sequences. *arXiv preprint arXiv:2201.11838*.

Lin, Y.-W., Zhou, Y., Faghri, F., Shaw, M. J., & Campbell, R. H. (2019). Analysis and prediction of unplanned intensive care unit readmission using recurrent neural networks with long short-term memory. *PloS one*, *14*(7), e0218942.

Lipscomb, C. E. (2000). Medical subject headings (mesh). *Bulletin of the Medical Library Association*, *88*(3), 265.

Liu, F., Shareghi, E., Meng, Z., Basaldella, M., & Collier, N. (2021, June). Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 4228–4238). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.naacl-main.334` doi: 10.18653/v1/2021.naacl-main.334

Liu, J., Chen, Y., Liu, K., Bi, W., & Liu, X. (2020, November). Event extraction as machine reading comprehension. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 1641–1651). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.emnlp-main.128` doi: 10.18653/v1/2020.emnlp-main.128

Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lo, K., Wang, L. L., Neumann, M., Kinney, R., & Weld, D. (2020, July). S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 4969–4983). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.acl-main.447` doi: 10.18653/v1/2020.acl-main.447

Lu, Q., de Silva, N., Dou, D., Nguyen, T. H., Sen, P., Reinwald, B., & Li, Y. (2020, December). Exploiting node content for multiview graph convolutional network and adversarial regularization. In *Proceedings of the 28th international conference on computational linguistics* (pp. 545–555). Barcelona, Spain (Online): International Committee on Computational Linguistics. Retrieved from https://aclanthology.org/2020.coling-main.47 doi: 10.18653/v1/2020.coling-main.47

Lu, Q., De Silva, N., Kafle, S., Cao, J., Dou, D., Nguyen, T. H., ... Li, Y. (2019). Learning electronic health records through hyperbolic embedding of medical ontologies. In *Proceedings of the 10th acm international conference on bioinformatics, computational biology and health informatics* (pp. 338–346).

Lu, Q., Dou, D., & Nguyen, T. (2022, December). ClinicalT5: A generative language model for clinical text. In *Findings of the association for computational linguistics: Emnlp 2022* (pp. 5436–5443). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.findings-emnlp.398

Lu, Q., Dou, D., & Nguyen, T. H. (2021a, November). Parameter-efficient domain knowledge integration from multiple sources for biomedical pre-trained language models. In *Findings of the association for computational linguistics: Emnlp 2021* (pp. 3855–3865). Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.findings-emnlp.325 doi: 10.18653/v1/2021.findings-emnlp.325

Lu, Q., Dou, D., & Nguyen, T. H. (2021b). Textual data augmentation for patient outcomes prediction. In *2021 ieee international conference on bioinformatics and biomedicine (bibm)* (pp. 2817–2821).

Lu, Q., & Du, Y. (2017). Wikipedia-based entity semantifying in open information extraction. In *2017 14th iapr international conference on document analysis and recognition (icdar)* (Vol. 1, pp. 765–770).

Lu, Q., Gurajada, S., Sen, P., Popa, L., Dou, D., & Nguyen, T. (2022, December). Cross-lingual short-text entity linking: Generating features for neuro-symbolic methods. In *Proceedings of the fourth workshop on data science with human-in-the-loop (language advances)* (pp. 8–14). Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.dash-1.2

Lu, Q., Nguyen, T. H., & Dou, D. (2021). Predicting patient readmission risk from medical text via knowledge graph enhanced multiview graph convolution. In *Proceedings of the 44th international acm sigir conference on research and development in information retrieval* (pp. 1990–1994).

Ma, X., Si, Y., Wang, Z., & Wang, Y. (2020). Length of stay prediction for icu patients using individualized single classification algorithm. *Computer methods and programs in biomedicine*, *186*, 105224.

Ma, X., Xu, P., Wang, Z., Nallapati, R., & Xiang, B. (2019). Domain adaptation with bert-based domain classification and data selection. In *Proceedings of the 2nd workshop on deep learning approaches for low-resource nlp (deeplo)* (pp. 76–83).

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Mascio, A., Kraljevic, Z., Bean, D., Dobson, R., Stewart, R., Bendayan, R., & Roberts, A. (2020, July). Comparative analysis of text classification approaches in electronic health records. In *Proceedings of the 19th sigbiomed workshop on biomedical language processing* (pp. 86–94). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.bionlp-1.9` doi: 10.18653/v1/2020.bionlp-1.9

McCray, A. T., Burgun, A., & Bodenreider, O. (2001). Aggregating umls semantic types for reducing conceptual complexity. *Studies in health technology and informatics*, *84*(0 1), 216.

McLaughlin, N., Del Rincon, J. M., & Miller, P. (2015). Data-augmentation for reducing dataset bias in person re-identification. In *2015 12th ieee international conference on advanced video and signal based surveillance (avss)* (pp. 1–6).

Menardi, G., & Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data mining and knowledge discovery*, *28*(1), 92–122.

Meng, Y., Speier, W., Ong, M. K., & Arnold, C. W. (2021). Bidirectional representation learning from transformers using multimodal electronic health record data to predict depression. *IEEE Journal of Biomedical and Health Informatics*, *25*(8), 3121–3129.

Michalopoulos, G., Wang, Y., Kaka, H., Chen, H., & Wong, A. (2020). Umlsbert: Clinical domain knowledge augmentation of contextual embeddings using the unified medical language system metathesaurus. *arXiv preprint arXiv:2010.10391*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *Interspeech* (Vol. 2, pp. 1045–1048).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2020). Deep learning based text classification: A comprehensive review. *arXiv preprint arXiv:2004.03705*.

Müller, M., Salathé, M., & Kummervold, P. E. (2020). Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *arXiv preprint arXiv:2005.07503*.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, *5*(1), 32–38.

Nan, G., Guo, Z., Sekulić, I., & Lu, W. (2020). Reasoning with latent structure refinement for document-level relation extraction. *arXiv preprint arXiv:2005.06312*.

Naseem, U., Dunn, A. G., Khushi, M., & Kim, J. (2022). Benchmarking for biomedical natural language processing tasks with a domain specific albert. *BMC bioinformatics*, *23*(1), 1–15.

Neumann, M., King, D., Beltagy, I., & Ammar, W. (2019, August). ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th bionlp workshop and shared task* (pp. 319–327). Florence, Italy: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/W19-5034` doi: 10.18653/v1/W19-5034

Nguyen, D. Q., Vu, T., & Tuan Nguyen, A. (2020, October). BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations* (pp. 9–14). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.emnlp-demos.2` doi: 10.18653/v1/2020.emnlp-demos.2

Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems* (pp. 6338–6347).

Nickel, M., Tresp, V., & Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *Icml* (Vol. 11, pp. 809–816).

Nikolentzos, G., Tixier, A., & Vazirgiannis, M. (2020). Message passing attention networks for document understanding. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 34, pp. 8544–8551).

Ou, M., Cui, P., Pei, J., Zhang, Z., & Zhu, W. (2016). Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1105–1114).

Ozyurt, I. B. (2020, November). On the effectiveness of small, discriminatively pre-trained language representation models for biomedical text mining. In *Proceedings of the first workshop on scholarly document processing* (pp. 104–112). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.sdp-1.12` doi: 10.18653/v1/2020.sdp-1.12

Pakhomov, S., McInnes, B., Adam, T., Liu, Y., Pedersen, T., & Melton, G. B. (2010). Semantic similarity and relatedness between clinical terms: an experimental study. In *Amia annual symposium proceedings* (Vol. 2010, p. 572).

Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., & Zhang, C. (2018). Adversarially regularized graph autoencoder for graph embedding. In *Proceedings of the 27th international joint conference on artificial intelligence* (pp. 2609–2615).

Papanikolaou, Y., & Pierleoni, A. (2020). Dare: Data augmented relation extraction with gpt-2. *ArXiv*, *abs/2004.13845*.

Park, J., Lee, M., Chang, H. J., Lee, K., & Choi, J. Y. (2019). Symmetric graph convolutional autoencoder for unsupervised graph representation learning. In *Proceedings of the ieee international conference on computer vision* (pp. 6519–6528).

Park, S., Bae, S., Kim, J., Kim, T., & Choi, E. (2022, 07–08 Apr). Graph-text multi-modal pre-training for medical representation learning. In G. Flores, G. H. Chen, T. Pollard, J. C. Ho, & T. Naumann (Eds.), *Proceedings of the conference on health, inference, and learning* (Vol. 174, pp. 261–281). PMLR. Retrieved from `https://proceedings.mlr.press/v174/park22a.html`

Peng, B., Zhu, C., Zeng, M., & Gao, J. (2020). Data augmentation for spoken language understanding via pretrained models. *arXiv e-prints*, arXiv–2004.

Peng, H., Li, J., He, Y., Liu, Y., Bao, M., Wang, L., ... Yang, Q. (2018). Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 world wide web conference* (pp. 1063–1072).

Peng, Y., Yan, S., & Lu, Z. (2019a). Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 18th bionlp workshop and shared task (bionlp)* (pp. 58–65).

Peng, Y., Yan, S., & Lu, Z. (2019b, August). Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th bionlp workshop and shared task* (pp. 58–65). Florence, Italy: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/W19-5006` doi: 10.18653/v1/W19-5006

Pennington, J., Socher, R., & Manning, C. (2014, October). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543). Doha, Qatar: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/D14-1162` doi: 10.3115/v1/D14-1162

Pereira, L., Liu, X., Cheng, F., Asahara, M., & Kobayashi, I. (2020). Adversarial training for commonsense inference. In *Proceedings of the 5th workshop on representation learning for nlp (repl4nlp)* (pp. 55–60).

Perotte, A., Pivovarov, R., Natarajan, K., Weiskopf, N., Wood, F., & Elhadad, N. (2013). Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, *21*(2), 231–237.

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining* (pp. 701–710).

Peters, M. E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019). 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)* (pp. 43–54).

Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., & Gurevych, I. (2021). AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics (eacl).* Association for Computational Linguistics.

Pfeiffer, J., Rücklé, A., Poth, C., Kamath, A., Vulić, I., Ruder, S., ... Gurevych, I. (2020). Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations (emnlp)* (pp. 46–54).

Phan, L. N., Anibal, J. T., Tran, H., Chanana, S., Bahadroglu, E., Peltekian, A., & Altan-Bonnet, G. (2021). *Scifive: a text-to-text transformer model for biomedical literature.*

Poerner, N., Waltinger, U., & Schütze, H. (2019). Bert is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised qa. *arXiv preprint arXiv:1911.03681*.

Ponzoni, C. R., Corrêa, T. D., Filho, R. R., Serpa Neto, A., Assunção, M. S., Pardini, A., & Schettino, G. P. (2017). Readmission to the intensive care unit: incidence, risk factors, resource use, and outcomes. a retrospective cohort study. *Annals of the American Thoracic Society*, *14*(8), 1312–1319.

Qi, W., Yan, Y., Gong, Y., Liu, D., Duan, N., Chen, J., ... Zhou, M. (2020, November). ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the association for computational linguistics: Emnlp 2020* (pp. 2401–2410). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.findings-emnlp.217` doi: 10.18653/v1/2020.findings-emnlp.217

Qiu, L., Xiao, Y., Qu, Y., Zhou, H., Li, L., Zhang, W., & Yu, Y. (2019). Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 6140–6150).

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, *63*(10), 1872–1897.

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE transactions on systems, man, and cybernetics*, *19*(1), 17–30.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., ... others (2021). Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(140), 1-67. Retrieved from `http://jmlr.org/papers/v21/20-074.html`

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., ... others (2018). Scalable and accurate deep learning with electronic health records. *NPJ digital medicine*, *1*(1), 1–10.

Ramponi, A., van der Goot, R., Lombardo, R., & Plank, B. (2020, November). Biomedical event extraction as sequence labeling. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 5357–5367). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.emnlp-main.431` doi: 10.18653/v1/2020.emnlp-main.431

Rasmy, L., Xiang, Y., Xie, Z., Tao, C., & Zhi, D. (2021). Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, *4*(1), 1–13.

Rebuffi, S.-A., Bilen, H., & Vedaldi, A. (2017). Learning multiple visual domains with residual adapters. In *Proceedings of the 31st international conference on neural information processing systems (neurips)* (pp. 506–516).

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on artificial intelligence-volume 1* (pp. 448–453).

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics (TACL)*, *8*, 842–866.

Romanov, A., & Shivade, C. (2018a). Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 conference on empirical methods in natural language processing (emnlp)* (pp. 1586–1596).

Romanov, A., & Shivade, C. (2018b, October-November). Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 1586–1596). Brussels, Belgium: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/D18-1187` doi: 10.18653/v1/D18-1187

Rücklé, A., Geigle, G., Glockner, M., Beck, T., Pfeiffer, J., Reimers, N., & Gurevych, I. (2020). Adapterdrop: On the efficiency of adapters in transformers. *arXiv preprint arXiv:2010.11918*.

Ruksakulpiwat, S., Kumar, A., & Ajibade, A. (2023). Using chatgpt in medical research: Current status and future directions. *Journal of Multidisciplinary Healthcare*, 1513–1520.

Rumshisky, A., Ghassemi, M., Naumann, T., Szolovits, P., Castro, V., McCoy, T., & Perlis, R. (2016). Predicting early psychiatric readmission with natural language processing of narrative discharge summaries. *Translational psychiatry*, *6*(10), e921.

Sachan, D., Patwary, M., Shoeybi, M., Kant, N., Ping, W., Hamilton, W. L., & Catanzaro, B. (2021, August). End-to-end training of neural retrievers for open-domain question answering. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)* (pp. 6648–6662). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.acl-long.519` doi: 10.18653/v1/2021.acl-long.519

Sala, F., De Sa, C., Gu, A., & Ré, C. (2018). Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning* (pp. 4457–4466).

Salton, G. (1991). Developments in automatic text retrieval. *science*, *253*(5023), 974–980.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513–523.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Schuster, M., & Nakajima, K. (2012). Japanese and korean voice search. In *2012 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 5149–5152).

Seco, N., Veale, T., & Hayes, J. (2004). An intrinsic information content metric for semantic similarity in wordnet. In *Ecai* (Vol. 16, p. 1089).

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., & Eliassi-Rad, T. (2008). Collective classification in network data. *AI magazine*, *29*(3), 93–93.

Sennrich, R., Haddow, B., & Birch, A. (2016, August). Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1715–1725). Berlin, Germany: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/P16-1162` doi: 10.18653/v1/P16-1162

Shang, J., Ma, T., Xiao, C., & Sun, J. (2019). Pre-training of graph augmented transformers for medication recommendation. *arXiv preprint arXiv:1906.00346*.

Shi, H., Fan, H., & Kwok, J. T. (2019). Effective decoding in graph auto-encoder using triadic closure. *arXiv preprint arXiv:1911.11322*.

Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Text data augmentation for deep learning. *Journal of big Data*, *8*(1), 1–34.

Sienčnik, S. K. (2015). Adapting word2vec to named entity recognition. In *Proceedings of the 20th nordic conference of computational linguistics (nodalida 2015)* (pp. 239–243).

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., ... others (2022). Large language models encode clinical knowledge. *arXiv preprint arXiv:2212.13138*.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., ... others (2023). Large language models encode clinical knowledge. *Nature*, 1–9.

Slee, V. N. (1978). The international classification of diseases: ninth revision (icd-9). *Annals of internal medicine*, *88*(3), 424–426.

Smith, K., Megyesi, B., Velupillai, S., & Kvist, M. (2014). Professional language in swedish clinical text: Linguistic characterization and comparative studies. *Nordic Journal of Linguistics*, *37*(2), 297–323.

Song, K., Tan, X., Qin, T., Lu, J., & Liu, T.-Y. (2019). Mass: Masked sequence to sequence pre-training for language generation. In *International conference on machine learning* (pp. 5926–5936).

Stearns, M. Q., Price, C., Spackman, K. A., & Wang, A. Y. (2001). Snomed clinical terms: overview of the development process and project status. In *Proceedings of the amia symposium* (p. 662).

Su, P., & Vijay-Shanker, K. (2020). Investigation of bert model on biomedical relation extraction based on revised fine-tuning mechanism. In *2020 ieee international conference on bioinformatics and biomedicine (bibm)* (pp. 2522–2529).

Sun, T., Shao, Y., Qiu, X., Guo, Q., Hu, Y., Huang, X., & Zhang, Z. (2020b, December). CoLAKE: Contextualized language and knowledge embedding. In *Proceedings of the 28th international conference on computational linguistics* (pp. 3660–3670). Barcelona, Spain (Online): International Committee on Computational Linguistics. Retrieved from `https://aclanthology.org/2020.coling-main.327` doi: 10.18653/v1/2020.coling-main.327

Sun, T., Shao, Y., Qiu, X., Guo, Q., Hu, Y., Huang, X.-J., & Zhang, Z. (2020a). Colake: Contextualized language and knowledge embedding. In *Proceedings of the 28th international conference on computational linguistics (coling)* (pp. 3660–3670).

Sun, Y., Wang, S., Feng, S., Ding, S., Pang, C., Shang, J., ... others (2021). Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Sushil, M., Ludwig, D., Butte, A. J., & Rudrapatna, V. A. (2022). Developing a general-purpose clinical language inference model from a large corpus of clinical notes. *arXiv preprint arXiv:2210.06566*.

Tang, L., & Liu, H. (2011). Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, *23*(3), 447–478.

Taylor, R., Kardas, M., Cucurull, G., Scialom, T., Hartshorn, A., Saravia, E., ... Stojnic, R. (2022). Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*.

Thillaisundaram, A., & Togia, T. (2019, November). Biomedical relation extraction with pre-trained language representations and minimal task-specific architecture. In *Proceedings of the 5th workshop on bionlp open shared tasks* (pp. 84–89). Hong Kong, China: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/D19-5713` doi: 10.18653/v1/D19-5713

Trieu, H.-L., Tran, T. T., Duong, K. N., Nguyen, A., Miwa, M., & Ananiadou, S. (2020). Deepeventmine: end-to-end neural nested event extraction from biomedical texts. *Bioinformatics*, *36*(19), 4910–4917.

Trouillon, T., Dance, C. R., Gaussier, É., Welbl, J., Riedel, S., & Bouchard, G. (2017). Knowledge graph completion via complex tensor factorization. *The Journal of Machine Learning Research*, *18*(1), 4735–4772.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., & Bouchard, G. (2016). Complex embeddings for simple link prediction. In *International conference on machine learning* (pp. 2071–2080).

Tu, T., Azizi, S., Driess, D., Schaekermann, M., Amin, M., Chang, P.-C., ... Natarajan, V. (2023). *Towards generalist biomedical ai.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st international conference on neural information processing systems (neurips)* (pp. 6000–6010).

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *stat*, *1050*, 20.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Veyseh, A. P. B., Lai, V., Dernoncourt, F., & Nguyen, T. H. (2021). Unleash gpt-2 power for event detection. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)* (pp. 6271–6282).

WA, K., EA, D., & DP, W. (1985). APACHE II: a severity of disease classification system. *Crit Care Med*, *13*(10), 818-829.

WA, K., JE, Z., DP, W., EA, D., & DE, L. (1981). Apache-acute physiology and chronic health evaluation: a physiologically based classification system. *Crit Care Med*, *9*(8), 591–597.

Wada, S., Takeda, T., Manabe, S., Konishi, S., Kamohara, J., & Matsumura, Y. (2020). Pre-training technique to localize medical bert and enhance biomedical bert. *arXiv preprint arXiv:2005.07202*.

Wang, B., Xie, Q., Pei, J., Tiwari, P., Li, Z., et al. (2021). Pre-trained language models in biomedical domain: A systematic survey. *arXiv preprint arXiv:2110.05006*.

Wang, D., Cui, P., & Zhu, W. (2016). Structural deep network embedding. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1225–1234).

Wang, H., Li, J., Wu, H., Hovy, E., & Sun, Y. (2022). Pre-trained language models and their applications. *Engineering*.

Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X., Cao, C., ... others (2020). K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., & Tang, J. (2021). Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics (TACL)*, *9*, 176–194.

Wang, Y., Afzal, N., Fu, S., Wang, L., Shen, F., Rastegar-Mojarad, M., & Liu, H. (2020). Medsts: a resource for clinical semantic textual similarity. *Language Resources and Evaluation*, *54*, 57–72.

Wang, Y., Huang, M., Zhu, X., & Zhao, L. (2016). Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 606–615).

Wei, C.-H., Peng, Y., Leaman, R., Davis, A. P., Mattingly, C. J., Li, J., ... Lu, Z. (2016). Assessing the state of the art in biomedical relation extraction: overview of the biocreative v chemical-disease relation (cdr) task. *Database*, *2016*.

Wen, A., Fu, S., Moon, S., El Wazir, M., Rosenbaum, A., Kaggal, V. C., ... Fan, J. (2019). Desiderata for delivering nlp to accelerate healthcare ai advancement and a mayo clinic nlp-as-a-service implementation. *NPJ digital medicine*, *2*(1), 1–7.

Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on association for computational linguistics* (pp. 133–138).

Xia, R., Pan, Y., Du, L., & Yin, J. (2014). Robust multi-view spectral clustering via low-rank and sparse decomposition. In *Twenty-eighth aaai conference on artificial intelligence.*

Xiong, C., Zhong, V., & Socher, R. (2017). Dcn+: Mixed objective and deep residual coattention for question answering. *arXiv preprint arXiv:1711.00106*.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., ... Raffel, C. (2021, June). mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 483–498). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.naacl-main.41` doi: 10.18653/v1/2021.naacl-main.41

Xue, Y., Klabjan, D., & Yuan, L. (2018). Predicting icu readmission using grouped physiological and medication trends. *Artificial intelligence in medicine*, 4.

Yang, B., Yih, W.-t., He, X., Gao, J., & Deng, L. (2014). Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Yang, C., Liu, Z., Zhao, D., Sun, M., & Chang, E. (2015). Network representation learning with rich text information. In *Twenty-fourth international joint conference on artificial intelligence.*

Yang, X., Bian, J., Hogan, W. R., & Wu, Y. (2020). Clinical concept extraction using transformers. *Journal of the American Medical Informatics Association*, *27*(12), 1935–1942.

Yang, X., Chen, A., PourNejatian, N., Shin, H. C., Smith, K. E., Parisien, C., ... others (2022). A large language model for electronic health records. *npj Digital Medicine*, *5*(1), 194.

Yang, Y., Malaviya, C., Fernandez, J., Swayamdipta, S., Le Bras, R., Wang, J.-P., ... Downey, D. (2020). G-daug: Generative data augmentation for commonsense reasoning. In *Proceedings of the 2020 conference on empirical methods in natural language processing: Findings* (pp. 1008–1025).

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, *32*.

Yao, L., Mao, C., & Luo, Y. (2019a). Graph convolutional networks for text classification. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 7370–7377).

Yao, L., Mao, C., & Luo, Y. (2019b). Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Yasunaga, M., Bosselut, A., Ren, H., Zhang, X., Manning, C. D., Liang, P., & Leskovec, J. (2022). Deep bidirectional language-knowledge graph pretraining. *arXiv preprint arXiv:2210.09338*.

Yasunaga, M., Leskovec, J., & Liang, P. (2022, May). LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 8003–8016). Dublin, Ireland: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2022.acl-long.551` doi: 10.18653/v1/2022.acl-long.551

Yuan, H., Yuan, Z., Gan, R., Zhang, J., Xie, Y., & Yu, S. (2022). Biobart: Pretraining and evaluation of a biomedical generative language model. *arXiv preprint arXiv:2204.03905*.

Yuan, Z., Liu, Y., Tan, C., Huang, S., & Huang, F. (2021, June). Improving biomedical pretrained language models with knowledge. In *Proceedings of the 20th workshop on biomedical language processing* (pp. 180–190). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.bionlp-1.20 doi: 10.18653/v1/2021.bionlp-1.20

Yuan, Z., Zhao, Z., Sun, H., Li, J., Wang, F., & Yu, S. (2022). Coder: Knowledge-infused cross-lingual medical term embedding for term normalization. *Journal of biomedical informatics*, *126*, 103983.

Yunxiang, L., Zihan, L., Kai, Z., Ruilong, D., & You, Z. (2023). Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge. *arXiv preprint arXiv:2303.14070*.

Zhang, D., Li, T., Zhang, H., & Yin, B. (2020). On data augmentation for extreme multi-label classification. *arXiv preprint arXiv:2009.10778*.

Zhang, K., Yu, J., Yan, Z., Liu, Y., Adhikarla, E., Fu, S., ... Sun, L. (2023). *Biomedgpt: A unified and generalist biomedical generative pre-trained transformer for vision, language, and multimodal tasks.*

Zhang, X., Dou, D., & Wu, J. (2020). Learning conceptual-contextual embeddings for medical text. In *Aaai* (pp. 9579–9586).

Zhang, Y., Chen, Q., Yang, Z., Lin, H., & Lu, Z. (2019). Biowordvec, improving biomedical word embeddings with subword information and mesh. *Scientific data*, *6*(1), 1–9.

Zhang, Y., Yu, X., Cui, Z., Wu, S., Wen, Z., & Wang, L. (2020). Every document owns its structure: Inductive text classification via graph neural networks. *arXiv preprint arXiv:2004.13826*.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th annual meeting of the association for computational linguistics (acl)* (pp. 1441–1451).

Zheng, S., Zhu, Z., Zhang, X., Liu, Z., Cheng, J., & Zhao, Y. (2020). Distribution-induced bidirectional generative adversarial network for graph representation learning. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 7224–7233).

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 207–212).

Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., & Chua, T.-S. (2021). Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.