Spring 2021

Natural Language Processing and Graph Representation Learning for Clinical Data

David Chang
Yale University Graduate School of Arts and Sciences, david.chang@yale.edu

Abstract

Natural Language Processing and Graph Representation Learning for Clinical Data

David Chang

2021

The past decade has witnessed remarkable progress in biomedical informatics and its related fields: the development of high-throughput technologies in genomics, the mass adoption of electronic health records systems, and the AI renaissance largely catalyzed by deep learning.

Deep learning has played an undeniably important role in our attempts to reduce the gap between the exponentially growing amount of biomedical data and our ability to make sense of them. In particular, the two main pillars of this dissertation—natural language processing and graph representation learning—have improved our capacity to learn useful representations of language and structured data to an extent previously considered unattainable in such a short time frame.

In the context of clinical data, characterized by its notorious heterogeneity and complexity, natural language processing and graph representation learning have begun to enrich our toolkits for making sense and making use of the wealth of biomedical data beyond rule-based systems or traditional regression techniques.

This dissertation comes at the cusp of such a paradigm shift, detailing my journey across the fields of biomedical and clinical informatics through the lens of natural language processing and graph representation learning. The takeaway is quite optimistic: despite the many layers of inefficiencies and challenges in the healthcare ecosystem, AI for healthcare is gearing up to transform the world in new and exciting ways.

Natural Language Processing and Graph Representation Learning for Clinical Data

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
David Chang

Dissertation Director: Cynthia Brandt and Richard Andrew Taylor

June 2021

Contents

List of Figures

List of Tables

Acknowledgements

An acknowledgment
is an acceptance of truth.
lux et veritas.

I would not be here
if it weren't for my parents
and their undying love.

PhD from Yale
sounds like something nice I can
put up on my wall.

Oreo, baby,
you are the best decision
I have ever made.

Five years is a long
time to spend waiting for the
next thing to come,

And now that it’s here,
I’m glad I spent all this time
becoming someone.

I would like to thank
my parents, my friends, my dog,
and everyone else.

Chapter 1

Introduction

Much of the progress in computational biology, bioinformatics, and medical informatics in the past two decades has been defined by our attempts to narrow the tremendous gap between the rapidly growing amount of data and our ability to make sense of them. Advances in next-generation sequencing technologies and the widespread adoption of electronic health record (EHR) systems, for instance, have led to massive accumulation and digitization of biomedical and clinical data that remain largely underutilized for research.

Driven by the exponential growth of data and compute power, the past decade has seen a remarkable boom in the field of deep learning (DL) and the resurgence of both academic and public interest in artificial intelligence (AI), which revolutionized many fields of study—most notably computer vision, reinforcement learning, and natural language processing (NLP). One way to get a sense of the progress and the growing level of research interest in AI would be to list a handful of notable and recognizable breakthrough moments just in the past three years: AlphaZero [146], BERT [55], GPT-3 [28], and most recently DALLIE[1] and AlphaFold[2]. Another way would be to look at the popularization of major AI and machine learning (ML) conferences such

[1] https://openai.com/blog/dall-e/

[2] https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology

as NeurIPS, ACL, EMNLP, ICLR, ICML, CVPR, and AAAI, which saw attendance and submissions grow exponentially each year over the past decade.

While progress across different domains can occur at various rates and under different constraints, there has been a unifying trend of a paradigm shift in machine learning away from systems that rely heavily on manual rule construction or feature engineering toward systems that automatically learn a hierarchy of features from the data. In essence, the focus of manual labor in informatics and machine learning has significantly shifted toward the design of model architectures and the training and evaluation of the models.

In the context of biomedical and health informatics, the success of deep learning has reaffirmed the feasibility of precision medicine and large-scale data analysis, ushering in wide-spread adoption of deep learning techniques on clinical and biomedical data with promising results [39]. A substantial portion of data pertaining to patients and their care is stored as free-text in clinical notes and reports, but much of the information encoded in them has been difficult to extract and has remained effectively out of reach for most practitioners and researchers without NLP expertise. This recognition, in light of recent progress in the field of NLP and the increasing accessibility of tools for implementing newer models, motivated a growing community of researchers in clinical NLP to develop methodologies to improve our capacity to extract and use clinically relevant information from text data, drawing on knowledge and innovations from fields spanning linguistics, NLP, data mining, and machine learning. While medical images are also an important modality of patient data that has received the lion's share of public attention, they are simply outside the scope of this dissertation.

1.1 Clinical natural language processing

The tremendous progress made in general domain NLP in the 2010s did not transfer immediately to clinical NLP. There is usually some lag for progress in general domain NLP/DL to "trickle down" to more specialized domains and applications due to various constraints and the initial scarcity of expertise and know-hows in the research community. Clinical NLP researchers were confronted with the reality of the unique challenges and obstacles in the clinical domain that prevented techniques developed in general domain NLP from being applied easily to clinical text.

One helpful way to conceptualize the differences across domains is within the framework of sublanguages [77]. Biomedical language can be thought of as a sublanguage of general English, for instance, and Friedman et al. further subcategorize biomedical language into biomolecular and clinical sublanguages, demonstrating that each has its own set of semantic classes, relations, and grammars that constrain its linguistic structure [62]. In addition, Huske-Kraus et al. provide a discussion of the data generation process in the clinical setting and an overview of the linguistic characteristics of clinical text that add layers of complexity in the context of natural language generation of clinical text [88]. Along similar lines, Feldman et al. show that there are considerable linguistic differences even across note categories in the MIMIC-III corpus [90] as an example of such complexity [58]. Aside from being a unique sublanguage with patient and disease-centered semantics and relations, clinical text also suffers from numerous issues that make its analysis even more difficult: incomplete sentences, incorrect grammar, omitted verbs and entities, ambiguous shorthand and abbreviations, misspellings, acronyms, jargon, negation, non-standard document structure, and more [50].

In addition to these challenges, the issue of data access and annotation (labeling) acted as a major obstacle to progress in clinical NLP. Chapman et el. identified several barriers in clinical NLP—lack of access to shared, annotated datasets for model training and benchmarking, lack of standards for annotations, and lack of reproducibility and collaboration—and argue that novel approaches to address those barriers are necessary to facilitate progress in the field [36]. A few years later, Savova et al. catalogued existing annotated clinical corpora, described progress made toward improving the quality of data annotation up to that point, and emphasized the need for more concerted efforts to construct and share high-quality clinical corpora [141]. Despite the awareness of such obstacles, and due to the difficulty of effectively surmounting them, the barrier to entry for researchers looking to break into clinical NLP remained high until around 2018, when the advent of transfer learning, among other factors, drastically lowered that barrier.

Another important concern with clinical text is patient privacy and the resulting strict limitations on data access–arguably the biggest obstacle in the field. In order for a researcher to work directly with clinical text, she has to be affiliated with the hospital or university and go through a nontrivial amount of training and paperwork to obtain access. This is why MIMIC-III [90] has become one of the most important and commonly used datasets in the field; it contains a wealth of data about ICU patients who stayed in the Beth Israel Deaconess Medical Center between 2001 and 2012, including their de-identified clinical notes. Due to technical and legal challenges, it is rare for institutions to decide to make public datasets containing patient data, especially clinical text. There are several avenues through which the field can overcome this: de-identification tools can be used to eliminate protected health information (PHI) from clinical notes and broaden access to those notes (as done with MIMIC-III), and federated learning [131] and sandbox projects can facilitate research projects without needing to transfer data outside designated servers. The recent partnership between Mayo Clinic and the biotech startup nference is an example where an academic medical center can partner with companies to make data usable for research with the help of de-identification tools and federated learning [116].

Despite all the problems with data quality and availability, clinical NLP shares a lot of the same building blocks as general NLP in terms of tasks and approaches. Some of the most common tasks in NLP are applicable in the clinical domain: part-of-speech tagging, dependency parsing, named entity recognition, relation extraction, coreference resolution, summarization, question answering, and natural language understanding. And although clinical NLP has been dominated by rule-based approaches or more traditional machine learning algorithms until recently [166], the past few years have seen growing adoption of deep learning and newer NLP techniques [139] [172]. More concretely, clinical information extraction papers published between 2009-2016 were mostly completely rule-based or used some variation of logistic regression, support vector machine, or random forests. And in a summary of clinical NLP systems submitted to shared tasks organized by i2b2/VA [155], Filannino et al. reported a clear trend in more data-driven methods over rule-based methods and reiterated the importance of annotated clinical corpora accessible to the research community to facilitate further progress [60].

A more recent review by Liu et al. provides a brief survey of recent applications of deep learning on EHR data, many of which fall under clinical NLP, and they note an emerging trend of deep learning models [108]. By the end of 2018, recurrent neural networks (RNNs) [138] and Word2vec [113] embeddings were the most popular methods in the literature used for information extraction tasks, with a long tail of other methods and specific tasks [172].

However, it wasn't until 2019 that a noticeable paradigm shift occurred in the clinical NLP literature: the supervised learning era that dominated the ML space up to the mid-2010s gave way to the era of transfer learning and self-supervised learning. The catalyst for this shift was the now-famous BERT [55] paper in 2018, which itself was built on a previous landmark paper from 2017 introducing the transformer model [158] with a highly parallelizable architecture that enabled efficient training of NLP models on large corpora of text (whereas previously, then-popular RNNs and LSTMs [82] were bottlenecked by their recurrence). BERT quickly rose to stardom after it demonstrated undeniably superior performance in numerous benchmark NLP tasks, motivating a whole slew of subsequent work and applications that have since dominated the NLP literature. The rise of transfer learning in NLP is thoroughly discussed in the 2019 PhD thesis by Sebastian Ruder, one of the leading young researchers in NLP with a focus on transfer learning and multilingual learning [136].

As previously mentioned, one noticeable impact of the rise of transfer learning and large transformer-based language models like BERT is that the barrier to entry for clinical NLP research has been lowered dramatically. The anxieties about not having publicly accessible clinical text datasets of high enough quality and quantity, which was essential for supervised learning, were assuaged by the availability of large pretrained language models which could easily be used out-of-the-box and fine-tuned to a domain-specific dataset of modest size to yield good results. It wasn't long before different research groups trained their own BERT models on clinical and biomedical text corpora, obtained significant improvements across domain-specific benchmark tasks to demonstrate that the benefits of transfer learning can indeed be reaped in the clinical and biomedical domains as well, and made their pretrained models publicly available, greatly facilitating productivity, interest, and research activity in the clinical NLP community [8] [139].

There are several important drivers of progress in the post-BERT era. First, the open-source culture embraced by the ML community and the remarkable evolution of open-source deep learning frameworks such as Tensorflow [1] and Pytorch [120], as well as libraries built on top of them (most notably Huggingface's Transformers library [171]), contributed greatly to the pace of research and the quality of published code. Second, the parallelizability of transformer-based language models and their capacity for transfer learning, along with the growth in compute power, enabled most researchers to relatively easily implement and experiment with state-of-the-art models. Third, the growth of top AI/ML conferences and workshops as legitimate venues for timely publication, networking, and feedback accelerated the flow of knowledge and insights. Two workshops in particular–BioNLP [53] and ClinicalNLP [139]–have been crucial hubs in the clinical NLP research community, and reading through their proceedings can be an excellent way to gauge the state of the field in a given year. The gradual acceptance of deep learning by the community, the increasing prevalence of transformer-based language models in the last two years, and the general improvement in the quality and quantity of submissions are some trends that can be gleaned from the proceedings.

The capacity of deep learning models to learn rich representations of data has not only transformed biomedical NLP but also motivated increasing efforts to develop integrative approaches to handle complex, heterogeneous data conducive to a systems view of patient-relevant data. Essentially, when the NLP pipeline is working well, one must start to wonder what to do about all the other types of data that's available and how to leverage the relational aspect of multiple modalities of data. And this line of inquiry lends itself naturally to the domain of graph representation learning.

1.2 Graph Representation Learning

Graph representation learning is a field that combines knowledge from traditional graph theory and network science with recent developments in machine learning to effectively learn with graph-structured data. In its current incarnation, graph representation learning is primarily concerned with approaches to incorporate graph data into the modern ML/DL pipeline. Graph representation learning can be broken down into two broad categories that have, in recent years, converged quite substantially: knowledge graph embeddings and graph neural networks.

Knowledge graph embeddings (KGEs) are methods developed mostly within the past decade that map entities and relations from knowledge graphs (i.e. collections of facts or triples) to an embedding space, analogous to word embeddings for text. Graph neural networks (GNNs), also referred to as graph convolutional networks (GCNs) [142] were largely motivated by attempts to generalize the notion of the convolution operator in convolutional neural networks (CNNs) [99] to the graph domain, and they can map graph-structured data to an embedding space. Both of these classes of models have gone from a niche topic to some of the most popular subfields in deep learning and AI conferences in the span of a decade.

The trajectory of graph representation learning from niche to mainstream can be traced with the following set of notable developments (not chronologically nor comprehensively): the translation-based knowledge graph embedding model called TransE [24] was presented at NIPS 2013, popularizing the methodology and motivating a whole collection of subsequent models; the 2016 publication of Kipf and Welling's Graph Convolutional Network paper [94], which simplified previous iterations of graph neural networks and provided an efficient implementation in Tensorflow, also popularized the field and motivated a whole collection of subsequent models; the 2018 "part-position paper, part review, and part unification" of relational inductive biases, deep learning, and graph networks [16] further stimulated interest in the field while providing valuable insights and a unifying framework for a scattered and newly emerging literature; the development of open-source libraries for graph-based deep learning such as Pytorch Geometric [59], Deep Graph Library (DGL) [162], and Graph Nets lowered the barrier for researchers and accelerated the pace of innovation; William Hamilton's 2020 book on graph representation learning [74] came out as one of the first textbooks for the field; the development of two libraries built on Pytorch Geometric and DGL–Benchmarking GNNs [57] and Open Graph Benchmark [168]–provided much-needed benchmarking and standardization in the field while encouraging more reproducible and interesting research; and the recent sequence of GNN and KGE related workshops at top AI/ML conferences have solidified their place as some of the most active and prolific subfields today.

Research efforts at the intersection of NLP and GRL naturally started to spring up, and there have been several lines of inquiry including the incorporation of KGs into the transformer-based language model (LM) pretraining pipeline (in the form of an auxiliary task in a multi-task learning setting, for instance), visual question answering (VQA) or multi-hop QA and reasoning tasks that are conducive to the sort of multimodal modeling enabled by graph-based techniques, and other methods using some combination of LMs and GCNs to jointly process text and relevant structured information (e.g. explicitly tagged entities accompanying sentences from wikipedia).

Callahan et al. [29] provide an excellent review of knowledge-based biomedical data science as of early 2020, touching on many past examples of KG-based applications in clinical and biological data. They have a very brief subsection on knowledge-based NLP applications, most of which focus on automatic KG construction methods and relatively simple information extraction tasks, and the brevity of this subsection is an indication of how underexplored the intersection is currently. Notable, they also recommend two specific areas that deserve more attention: learning biomedical concept embeddings and integrating biomedical KGs into NLP applications, both of which are explored throughout this dissertation.

Michael Galkin's series of blog posts summarizing notable sets of papers presented at major AI conferences and workshops in the past two years offer a great overview of the kinds of topics and methods that have emerged at the intersection of NLP and GRL. But, given the novelty of this subfield, the methodologies are understandably still far from mature, and much work remains to be done to fully realize the potential of leveraging the best aspects of the two fields to effectively combine multiple modalities of data.

In this chapter, I gave a broad overview of the fields of NLP and GRL and their burgeoning intersection, and Chapter 2 provides a more in-depth background on the methods and applications in those areas. The remaining chapters document my journey across these topics within the clinical domain, starting with applying language models to chief complaint data from the emergency department (Chapter 3), moving over to training biomedical knowledge graph embeddings (Chapter 4), integrating them in a clinical semantic textual similarity task (Chapter 5), and finally introducing a more sophisticated methodological contribution that ties together recent advances in NLP and GRL to combine text and diagnosis codes for medical prediction tasks (Chapter 6). Chapter 7 concludes this dissertation with a summary of my work during my PhD.

Chapter 2

Background

This chapter provides relevant background on the fields of natural language processing (NLP) and graph representation learning (GRL), as well as recent developments in their intersection with a focus on applications to biomedical and clinical data. Far from being a comprehensive survey of the fields of interest, the following sections focus on topics and methods that are salient within the context of this dissertation. More specifically, in section 2.1, I give an overview of representation learning in NLP and the history of word representation, going into detail on three major modeling breakthroughs: RNNs, Transformers, and BERT. In section 2.2, I briefly summarize the field of GRL and its two main methodological directions, discussing some of the representative methods for each. Section 2.3 describes attempts made so far to combine the fields of NLP and GRL with biomedical and clinical data.

2.1 Natural Language Processing

One of the key aspects of a good machine learning or NLP model is the ability to learn an effective representation of the data. Particularly in NLP, the conversion of text from a symbolic, discrete form to a numerical representation is a necessary step in order to apply machine learning techniques on language data. Current deep learning models in NLP can be decomposed into several modular abstractions that perform some function. The following are modular abstractions that appear in many deep learning NLP models:

- Indexer: determines what a token is (e.g. word, word-pieces, or character), and how to convert tokens to integers with a mapping.
- Embedder: maps a sequence of token ids produced by the indexer to a sequence of vectors.
- Encoder: encodes input embeddings using a specified model to obtain a final representation of the input.
- Decoder: decodes the representation obtained through the encoder and generates output for evaluation.

These abstractions are meant to provide a framework in which to consider an NLP model, and the distinctions between them can be quite blurry. For instance, while a model like Word2vec [113] that acts as a lookup table for the indices of tokens can serve as an embedder, it can also be considered an encoder. In the case of machine translation using transformers [158], the transformer module acts as both the encoder and the decoder. But for a sentence classification task using a transformer, the decoder can simply be a fully connected output layer that computes the class label. The important thing is for there to be a mapping from the discrete, symbolic representation of the text (e.g. words or characters) to a dense numerical vector representation, with some transformations along the way, that can be manipulated within the deep learning framework.

2.1.1 Word embeddings

These dense vector representations of words are called word embeddings (since the models embed the words from the vocabulary to a vector space), and they were mainly popularized around 2013 by Word2vec and GloVe [122], which effectively trained a lookup table as a language model on large amounts of unlabeled text. Word2vec and subsequent word embedding models brought mainstream attention to the field of NLP by demonstrating that publicly available pretrained word embeddings could be used to boost performance generally across NLP applications. This was really the precursor to the era of transfer learning in NLP, a learning paradigm in which information from previously trained models can be leveraged to boost performance, accelerate training, and increase data efficiency in a different setting (i.e. in a different task or domain) with some fine-tuning on a smaller, task-specific dataset. The ability to simply download pretrained word embeddings from a server hosted by Google or Stanford and plug them into a generic machine learning model to obtain significant improvements was groundbreaking for NLP practitioners and researchers. And the showcasing of the capacity of these word embeddings to capture the semantic information in a language (as often demonstrated with the man:king::woman:queen analogy plotted as vectors and translations in the embedding space or neat, interactive visualizations of word embeddings[1]) contributed to the rise of NLP as one of the major pillars of deep learning alongside computer vision.

These pretrained word embeddings were frequently used as initial representations of words to be passed into recurrent neural networks (RNNs). RNNs are a family of deep learning models designed to work with sequential inputs such as sentences [138]. The main difference between RNNs and simple feed-forward neural networks is that RNNs have a connection between the previous state and the current state (i.e. recurrence), parameterized by the recurrent layer weight matrix. The model is

[1] https://projector.tensorflow.org/

formulated as follows:

$$h(t) = f(Ux(t) + Wh(t-1)), \tag{2.1}$$

where $x(t)$ is the input at time-step t (i.e. t-th word in sentence x) and U and W are the weight matrices to be learned during training. $h(t)$ is the model hidden state at time-step t and is a nonlinear function of $x(t)$ and $h(t-1)$, the model hidden state from the previous step. f is an activation function or a nonlinearity, typically a sigmoid fuction or ReLU [117], that is applied to the combined linear transformations of the input and the previous hidden state. The weight matrices U and W are a form of temporal weight sharing across time steps or layers, and this weight sharing helps encode translation invariance across time, allowing the model to extract features regardless of where they are in the sequence.

Once the model steps through all the inputs, the final hidden state can be used for the downstream task. For tasks that require an output at each time step as in text generation, the hidden state at each time step would be passed to an output layer to generate the output token after each step. For a classification task, the final hidden state is passed through an output layer to generate the predicted label. For a sequence-to-sequence task as in machine translation, the final hidden state may be used as the input for a decoder RNN.

Theoretically, RNNs are able to capture long-term dependencies among input features in a sequence by keeping a memory based on previous time steps through a recurrent process. However, vanilla RNNs can be very unstable during training due to the vanishing or exploding gradient problem, caused by the repeated multiplication of the recurrent weight matrix with the consequence of model states often converging to 0 or diverging to a large number [81]. In practice, a variant of RNNs called Long Short Term Memory (LSTM) networks [82] are most commonly used as a representative of the class of RNN models. LSTMs are just RNNs with a different update rule that was explicitly designed to better capture long-term dependencies among input features with more stable training dynamics. More specifically, an LSTM cell employs what are called gate mechanisms to control the flow of information from one layer to the next, allowing a more flexible update rule with an additive component, in contrast to the simply repeated multiplication by a weight matrix in a vanilla RNN. The memory cell in an LSTM is used to regulate the flow of information from the previous and current inputs, and the input gate, forget gate, and output gate control the proportions of information that flows from one step to the next.

Furthermore, Graves et al. proposed a bidirectional LSTM (bi-LSTM) [66], which simply applies an LSTM in both directions of the input sentence and concatenates the resulting hidden state vectors from each direction to get the final hidden state vector. Bi-LSTMs empirically produce better results by leveraging information flowing in both directions of the input sequence and have become a common method of processing sequential data in deep learning.

Even with more sophisticated sequence models, one of the main limitations of early word embeddings like Word2vec was that they produced a single static, context-insensitive embedding for each word and thus were unable to take into consideration the fact that the meaning of a word can change significantly depending on its context. For example, in the two snippets of text “open a bank account” and “on the river bank”, the word “bank” has a clearly different meaning in each context. Methods that attempted to address this shortcoming and to incorporate contextual information into word embeddings were termed contextual embeddings, and this line of research dominated the NLP space around 2016-2018 along with research on variants of RNN architectures.

In 2017, Peters et al. introduced Embeddings from Language Models (ELMo) [124], a new type of deep contextualized word representation that could model not only the syntax and semantics of words but also how word use varies across contexts (i.e. polysemy). ELMo is basically a stacked bi-LSTM trained on a large corpus of text, and the word embeddings are generated using the learned weights of the model. In contrast to previous word embedding models, ELMo embeddings are a function of the entire context of a word (i.e. the whole input sentence) and not merely a lookup table of individual words. Hence, ELMo introduced an approach to learn high-quality context-dependent word representations, advancing the state-of-the-art on numerous NLP benchmark tasks, notably the GLUE benchmark [161]. All subsequent state-of-the-art models beyond ELMo are contextual embeddings, most of which are based on the transformer model.

2.1.2 Transformers

In 2017, Vaswani et al. [158] produced state-of-the-art performance in machine translation in a paper titled "Attention Is All You Need". The title refers to the fact that the transformer model consists only of stacked attention layers and does away with the recurrent modules that were popular at the time. The transformer model allowed much more efficient parallel processing of tokens (leveraging increasing compute power from GPUs) and better modeling of long-range dependencies. It offered an attention-only approach to sequence modeling that addressed some of the major shortcomings of recurrent networks, including the information bottleneck and poor scaling and parallelizability.

The crux of the transformer model is the multi-headed self-attention mechanism: a scaled dot-product attention acting on a set of vectors:

$$Attention(Q, K, V) = softmax\Big(\frac{QK^T}{\sqrt{d}}\Big)V \tag{2.2}$$

The tensors Q (queries), K (keys), and V (values) are separate linear transformations of the same input tensor, parametrized by weight matrices W_Q, W_K, W_V, respectively. The term inside the parentheses on the right hand side of equation (2.2) quantifies the information overlap of the queries (Q) with the keys (K) corresponding to the values (V). Hence, the final attention matrix yielded by the equation is a linear combination of the value vectors weighted by the scaled dot-product of the keys and queries. The attention matrix is essentially a transformation of the input embeddings based on the inter-relatedness of those input embeddings.

While the self-attention mechanism is the defining component of the transformer model, other components are also important for the model to work in practice. The multi-headedness of the multi-headed self-attention comes from the fact that the dimensions of the attention module are chunked into multiple attention operations running in parallel. For instance, an attention module of dimension 768 would have 12 attention heads of dimension 64 each, and the results of the parallel operations would be concatenated to yield the final output of the attention module. Using multiple attention heads allows diverse couplings of queries and keys derived from the input vectors to calculate the attention coefficients while also providing multiple initialization points in the subspaces to reduce the chances of being stuck in bad local minima during training.

After the self-attention module, there is also a positional feed-forward network consisting of two linear layers with an activation function, the purpose of which is to increase model capacity and help warp the latent representations by "blowing up" the hidden states to a much higher dimension (typically x4) and then bringing them back to a lower dimension. A recent paper by Geva et al. [63] explores the feed-forward layers of the transformer model in depth, demonstrating that they operate as key-value memories that often learn human-interpretable patterns and help concentrate the probability mass in the output distributions on tokens that are likely to appear. The residual connections [75] in the transformer layers complement the feed-forward layers by refining their outputs to produce the final output distribution that effectively composes information from within and across the transformer layers.

Further, layer normalization [11] is used to control the norm of the hidden states and stabilize training, and a separate position embedding is trained on the positions of the tokens and added to the embeddings of the input tokens, since the self-attention mechanism is permutation invariant and doesn't explicitly encode the positional information in the sequence.

Overall, the transformer model combined a variety of tricks and tools available at the time in NLP in an innovative and effective way and steered the field in a direction away from recurrence-based models and toward more parallelizable and larger models. Devlin et al. leveraged a combination of the transformer model, large-scale training, and transfer learning to create BERT, profoundly changing the field of NLP for years to come.

2.1.3 BERT

In 2018, Devlin et al. introduced Bidirectional Encoder Representation from Transformers (BERT) [55], a new language representation model that advanced the state-of-the-art in NLP by a significant margin. There are several key features of the BERT model and training procedure that explain its effectiveness.

First, it broke away from the convention of unidirectional language models and enabled bidirectional language modeling by introducing a new training strategy called masked language modeling (MLM). The MLM training involves corrupting 15% of the sequence and training the model to correctly predict those corrupted tokens. The 15% of corrupted tokens are replaced with a "$[MASK]$" special token 80% of the time, replaced with a random word from the vocabulary 10% of the time, and kept the same the remaining 10% of the time. Besides the MLM pretraining task, BERT also has a next sentence prediction task, but this auxiliary task has later been shown to be unimportant [111].

Second, BERT uses a tokenizer and a vocabulary trained using word pieces as the tokens (subword tokens), fixing the vocabulary size at a manageable 30k while mitigating the out-of-vocabulary problem. Third, BERT uses an embedding of the tokens that consists of three separate embeddings: token embeddings (word embeddings for the tokens in the vocabulary, as usual), segment embeddings (a list of 0's and/or 1's depending on whether the example consists of one or two sentence segments), and position embeddings (introduced in the transformer paper). These three types of embeddings are summed to produce an informative representation of the input tokens. Fourth, BERT was trained on a large corpus consisting of 2.5 billion words from Wikipedia and 800 million words from the BookCorpus [190]. Fifth, BERT had the largest model architecture at the time of release with 335 million parameters for BERT-large and 110 million parameters for BERT-base. Lastly, a combination of BERT's architecture and training strategy allowed it to be fine-tuned to a wide variety of NLP tasks (sentence- or sentence-pair classification, multiple choice questions, question answering, text generation, sequence tagging, etc) and resulted in the best demonstration of the potential of transfer learning in NLP.

This advancement of the field was most apparent in the GLUE benchmark [161], a collection of nine sentence- or sentence-pair language understanding tasks built on existing datasets in NLP and selected to cover a wide range of difficulty and text genres. In the results table, the authors show that BERT was able to obtain several points of absolute improvement across 11 NLP tasks, an unprecedented accomplishment for a single model. Notably, fine-tuning BERT yielded large improvements in tasks with relatively small datasets, further proving BERT's effectiveness and that of transfer learning in general.

Thanks to the authors' efforts to open-source release a clean, well-documented, and usable code along with the pretrained BERT models, it quickly garnered a tremendous amount of attention of saw widespread use. Among a myriad of downstream effects this release has had, most notably it gave rise to the ‘pytorch-pretrained-bert’ package (now called Transformers [171]) open-sourced by a young startup called Huggingface, which has since become an essential player in the NLP space with the most actively developed and widely used library in the field.

BERT’s impact was so pervasive that it was granted its own subject area termed Bertology, which entailed research efforts attempting to explain BERT’s inner workings, improving on its efficiency, and innovating along different aspects of the model. Training increasingly larger language models on larger text corpora became a trend, sometimes reaching billions in the number of parameters [130]. As subsequent models kept iterating and improving upon the initial BERT release, the human performance for the GLUE benchmark was surpassed quickly, making it obsolete. An improved and more difficult benchmark called SuperGLUE [160] was released, but that has also now become obsolete since recently DeBERTa [76] and T5 surpassed human performance. The fact that two benchmarks of considerable difficulty and investment of resources have been made obsolete in the span of 2 years is a testament to the pace of progress in the field.

2.2 Graph Representation Learning

The field of graph representation learning grew rapidly alongside NLP in the past few years, inheriting a lot of the techniques and insights generated from deep learning. With the ever-increasing size and complexity of datasets available for analysis in recent decades, it became important to take on the challenge of leveraging machine learning to effectively learn the representations of graph-structured data.

It is useful to think about deep learning models—and machine learning algorithms in general—in terms of inductive biases (also known as learning bias). Inductive biases are a set of assumptions or constraints a learner can use to prioritize the search for better or more generalizable solutions. For example, L2 regularization is a form of inductive bias that prioritizes smaller parameter values and unique solutions. Also, popular deep learning model architectures such as convolutional neural networks or recurrent neural networks can be thought of as inducing specific relational inductive biases in the form of locality invariance and temporal invariance, respectively. The transformer model and its self-attention mechanism discussed above induce permutation invariance. In part, progress in deep learning can be described in terms of the importance of integrating assumptions or priors about the inherent structure of the data into the model architecture.

Graphs provide a natural generalization of different types of data such as images, text, networks, and so on. Hence, graphs are intuitively a promising way to approach the study of relational or structural inductive biases in our learning algorithms and data. Battaglia et al. explore the idea of relational inductive biases in the context of deep learning and advocate for a graph-based approach to inducing relational inductive biases in order to facilitate learning about entities and their relations. The part-review and part-unification paper also proposes a unifying framework for graph networks that can induce arbitrary types of relational inductive bias, providing a direction for more sophisticated, interpretable, and flexible patterns of reasoning [16].

A graph (G) can be defined as a set of nodes (V) and edges (E) between those nodes, and the structure of the graph can be represented by its adjacency matrix $A \in \mathbb{R}^{|V| \times |V|}$. The values in the adjacency matrix can be binary (values in $\{0, 1\}$), in which case the adjacency matrix denotes whether the nodes are connected or not connected; directed (values in $\{-1, 0, 1\}$, in which case A will not necessarily be symmetric; and weighted (values are continuous), in which case the elements of the adjacency matrix can represent the strength of connectivity between nodes. The nodes and edges in a graph can also be of multiple types, giving rise to heterogeneous or multi-relational graphs.

Traditional approaches to representing graph-structured data relied heavily on simple graph statistics and kernels to extract features for downstream tasks. However, these approaches were limited by the need for hand-engineered statistics and features. Recent advances in machine learning and deep learning gave rise to an alternative approach for handling graphs that is based on learned representations (hence graph representation learning), mirroring the paradigmatic shift witnessed in other fields like computer vision and NLP. The crux of this approach involves methods for learning node embeddings, a mapping from the graph-structured data to a dense vector space that encodes structural and semantic information about the graph. While there have been many types of methods for learning node embeddings, the rest of this section focuses on two major research directions in the field: graph neural networks and knowledge graph embeddings.

2.2.1 Graph neural network

Graph neural networks (GNNs), originally proposed by Merkwirth and Lengauer [112] and Scarselli et al. [143], have been steadily gaining popularity and research momentum, bolstered in recent years by the introduction of a simplified graph convolutional network and subsequent models leveraging techniques from other areas of deep learning [173] [186] [188] [26].

Graph convolution can be seen as a generalization of the convolution operator in a CNN; just as convolution layers in a CNN learn higher-level representations of images by gathering information from patches of pixels (which can be seen as a graph laid out in a grid-like structure), a graph convolutional layer obtains higher-level representations of the nodes in a graph by gathering information from their neighbors. The main distinction between convolutional layers in CNNs and graph convolutional layers is that while regular convolution only needs to operate over fixed connectivity patterns (patch of pixels that always have the same “connectivity” as a grid), graph convolution must be able to operate on arbitrary graphs with variable sizes and connectivity patterns. In other words, graph convolution requires permutation invariance: the results of the operation cannot depend on the arbitrary ordering of the nodes in the adjacency matrix.

A nice treatment of the generalization of convolution to graphs can be found in Kipf and Welling's popular GCN paper [93]. More specifically, one can consider graph convolution under the conceptual framework of message-passing networks [64] in which the representation of a node is iteratively updated using information (message) passed from its neighboring nodes. This can be simply expressed as the following equation:

$$H' = \sigma(AHW), \tag{2.3}$$

where W is the trainable weight matrix of the graph convolution layer, H is the previous hidden state of the nodes, H' is the updated hidden state, A is the adjacency matrix of the graph, and σ is a nonlinear activation function. In order to make it work in practice, a couple of tricks are used: adding self-loops to the adjacency matrix and normalizing the adjacency matrix with its degree matrix, as in the following formulation of GCN:

$$H' = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}HW), \tag{2.4}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix with added self connections and D is the degree matrix of $\tilde{A}$. It is worth noting that while the symmetric normalization trick in equation (2.4) was introduced in the original GCN paper by Kipf and Welling, proper normalization should be considered on a task by task basis since it can also lead to a loss of information. Usually, normalization by degrees is appropriate for tasks in which the node feature information is more useful than the structural information in the graph, or when a wide range of values for node degrees can lead to instabilities in training.

One of the most fascinating connections between NLP and GRL is the equivalence between the transformer model and a GNN model with multi-headed attention, introduced by Velickovic et al. as the Graph Attention Network (GAT) [159]. In essence, a regular feed-forward neural network can be seen as a special case of a graph neural network with a fully connected graph in which each layer aggregates information from all the nodes in the previous layer. A natural extension to this idea is the ability to flexibly attribute weights to the nodes in the input as a function of the node features using the attention mechanism. In the context of NLP, this was achieved by the development of the transformer module, which can be seen as a special formulation of a graph neural network with multi-headed self-attention.

An important issue with GNNs is over-smoothing: after several layers of message passing in graph convolution, the representations for the nodes can become too similar and uninformative. This is one of the reasons why stacking many layers to build deep GNN models is difficult; long-term dependencies in the graph would get lost over the iterations from over-smoothing. Insights shared by Xu et al. [174] are useful for conceptualizing this phenomenon. When using a k-layer GCN model, as k gets larger, the influence of every node to the final representation converges to the stationary distribution of random walks over the graph. This is especially problematic for real-world graphs that contain high-degree nodes, where node representations can more quickly approach an almost uniform distribution with more layers. So it was observed that naively building deeper GNN models actually led to loss of information about neighborhood structures and hurt performance. Using skip-connections (as popularized by residual connections in CNNs [75]) is one simple way to mitigate the problem of over-smoothing. Other approaches include gated updates [106], analogous to gated updates in variants of RNNs like LSTMs or the gated recurrent unit [40], and jumping knowledge connections [175].

While GNNs as a topic in GRL have grown at an unwieldy pace at times, with conference proceedings and online archives littered with papers describing proposals for novel components or models with little to no theoretical underpinnings or clear practical gains in performance, there have been increasing efforts to bolster the theoretical and practical foundations of research in the field. Open-source libraries, namely Pytorch Geometric[2] and Deep Graph Library (DGL)[3] built on top of popular deep learning frameworks have become an essential part of the community that helps establish best practices, promulgate new ideas, and accelerate research. More recently, repositories such as Benchmarking GNNs[4] and Open Graph Benchmark[5], both built using Pytorch, Pytorch Geometric, and DGL, have further contributed to the standardization and organization of knowledge in the field through a wide range of benchmark datasets, models, examples, and tasks.

As of early 2021, GNNs have solidified their place as one of the most actively growing and popular subfields of deep learning, and their effectiveness and flexibility have led to applications in a wide range of domains and problems.

2.2.2 Knowledge graph embeddings

Knowledge graphs and knowledge bases have long been important topics in AI and have seen a resurgence in popularity in the age of big data as the size and complexity of digitally stored knowledge grew substantially. While knowledge engineering and ontology construction are subjects with decades of history, it wasn't until the emergence of the field of GRL that modern machine learning and deep learning techniques were used to effectively leverage the vast amount of knowledge stored in these knowledge bases beyond writing queries and designing heavily hand-engineered methods to extract information from them. With many of today's leading AI companies (Amazon,

[2] https://github.com/rusty1s/pytorch_geometric

[3] https://github.com/dmlc/dgl

[4] https://github.com/graphdeeplearning/benchmarking-gnns

[5] https://github.com/snap-stanford/ogb

Google, Microsoft, Facebook, etc) using knowledge graphs in their technology, as well as the rapid evolution of methods for learning knowledge graph embeddings in the past decade, it is not surprising that knowledge graphs have become one of the most popular topics in all of the top AI conferences in the past couple of years.

Most KGE methods are designed with the knowledge graph completion task in mind. Given a multi-relational graph $G = (V, E)$ with edges defined as tuples of the form $e = (h, r, t)$ asserting that the head entity h has relation r to tail entity t, the goal of knowledge graph completion is to train a model that can predict missing edges in the knowledge graph (i.e. to predict r given $(h, ?, t)$). The trained model can also be used to predict the head or tail entities given the other two components, predict the likelihood of a given triple being a fact (triple classification), or predict the class to which an entity belongs (entity classification).

The problem of link prediction can be formulated as a reconstruction task: given the embeddings of two entities, try to reconstruct (predict) the correct relation type between them. One of the main distinctions between KGEs and node embedding methods designed for simple graphs (with one or very few relation types) is that the relation types are given explicit representation in the KGE models (i.e. relation types are represented as embeddings just like entities). Hence, the decoder or predictor component of the KGE model would take in a triple (a pair of entity embeddings and a relation type embedding) and yield a score that indicates the likelihood that the input triple is a fact.

The two main ingredients or distinguishing components for a lot of KGE methods are the decoder, also called the scoring function, and the loss function. Two of the most commonly used loss functions for KGE are cross-entropy with negative sampling and max-margin.

Cross-entropy with negative sampling is derived from the standard binary cross-entropy loss and is defined as:

$$L = \sum_{(h,r,t)\in E} \Big(-log(\sigma(s(e_h, e_r, e_t))) - \sum_{t_n \in P_{n,h}} [log(\sigma(-s(e_h, e_r, e_{t_n})))] \Big), \tag{2.5}$$

where σ is the logistic function, $P_{n,h}$ is a set of negative samples, and s is the scoring function or the decoder. The first term inside the outer summation is the log-likelihood that the positive triple is correctly predicted to be factual, and the second term is the expected log-likelihood that the model correctly predicted false for negative samples (corrupted triples not found in the knowledge graph). While in equation (2.5) the negative sampling is performed only on the tail entity, in practice negative samples are drawn from both the head and tail entities. Given that negative sampling strategies can significantly influence model training and performance, many approaches have been proposed in the literature, ranging from uniform sampling to more sophisticated methods that incorporate relation type constraints or adversarial training [4].

The max-margin loss, a common alternative to cross-entropy with negative sampling, is defined as:

$$L = \sum_{(h,r,t)\in E} \sum_{t_n \in P_{n,h}} max(0, -s(e_h, e_r, e_t) + s(e_h, e_r, e_{t_n}) + \Delta), \tag{2.6}$$

where Δ is the margin hyperparameter that controls how much we allow the model to be inaccurate. The score for a positive triple is compared to the score for a negative sample, and the loss equals 0 if the difference in the scores is greater than or equal to the margin. Max-margin loss is also called the hinge loss.

The other axis of variation in KGE methods, and the one that is most highly prioritized in publication, is the scoring function. Most models fall under the categories of bilinear models, tensor factorization models, and translation-based models. An early example of a KGE model is RESCAL [119], which represents relations as a bilinear product between subject and object entity vectors. Although a very expressive model, RESCAL is prone to overfitting due to the large number of parameters in the full rank relation matrix, increasing quadratically with the number of relations in the graph.

DistMult [178] is a special case of RESCAL with a diagonal matrix per relation, reducing overfitting. However, by limiting linear transformations on entity embeddings to a stretch, DistMult cannot model asymmetric relations. ComplEx [154] extends DistMult to the complex domain, enabling it to model asymmetric relations by introducing complex conjugate operations into the scoring function. SimplE [91] modifies Canonical Polyadic (CP) decomposition [80] to allow two embeddings for each entity (head and tail) to be learned dependently.

A recent model TuckER [14] is shown to be a fully expressive, linear model that subsumes several tensor factorization based approaches including all models described above.

TransE [24] is an example of an alternative *translational* family of KGE models, which regard a relation as a translation (vector offset) from the subject to the object entity vectors. Translational models have an additive component in the scoring function, in contrast to the multiplicative scoring functions of bilinear models. RotatE [150] extends the notion of translation to rotation in the complex plane, enabling the modeling of symmetry/antisymmetry, inversion, and composition patterns in knowledge graph relations.

While there have been many subsequent proposals for new KGE models, it has been very challenging to assess the validity of the individual results and the progress in the field due to a number of reasons: the lack of standardization and specification of experimental configurations and evaluation protocols [137], the problem of calibrating the model outputs [140], negative sampling strategies, the shortcomings of the most commonly used benchmark datasets (FB-15 and WN18 [24]), and lack of actively maintained and developed central hub for KGE implementations. But there have also been encouraging amounts of effort and attention placed on addressing those issues, with papers addressing the issues of model calibration [140], theoretical connections to word embeddings and their implication on how to interpret KGEs [7], efficient ways to train hyperbolic KGEs [32], and more diverse benchmark datasets and open-source libraries for KGE training [5] [27] [187].

2.3 Combining NLP, GRL, and clinical data

Shortly after BERT's publication and open-source release, the availability of pretrained BERT models along with well-documented scripts to continue training on a custom corpus in both Tensorflow and Pytorch (thanks to Huggingface's timely efforts) led to the appearance of several domain-specific versions of BERT models trained on biomedical and clinical text. Most of these models use general-domain pretrained models (e.g. BERT-base) as the starting point (also called a checkpoint) and further pretrain them on publicly available domain-specific text corpora, typically some combination of PubMed abstract[6], PubMed Central full-text[7], and MIMIC-III clinical notes [90].

BioBERT [101] is among the first of these domain-specific models that further pretrained BERT-base on the PubMed and PMC corpora. ClinicalBERT [8] by Alsentzer et al. came out shortly after, using the BioBERT model as the initialization checkpoint and further pretraining on the MIMIC-III corpus. BlueBERT [121], similar to ClinicalBERT, trained both BERT-base and BERT-large models on the PubMed and MIMIC-III corpora, simultaneously providing the Biomedical Language Understanding Evaluation (BLUE) benchmark with a collection of standard benchmark datasets in the biomedical domain.

Continuing on this trend, the recent proceedings of the 3rd Clinical NLP 2020

[6] https://pubmed.ncbi.nlm.nih.gov/

[7] https://www.ncbi.nlm.nih.gov/pmc/

workshop [139] featured several papers exploring the training of domain-specific language models on clinical text. MeDAL [169] further pretrained an Electra [47] model on a newly constructed large dataset for medical abbreviation disambiguation and demonstrated the usefulness of the dataset in the context of both pretraining and fine-tuning on downstream tasks. MS-BERT [51] trains a BERT-base model on a custom de-identified corpus of multiple sclerosis (MS) consult notes, providing the first publicly available transformer-based language model that isn't trained on the MIMIC-III corpus. Clinical XLNet [87] uses XLNet [179] as the initialization checkpoint and further pretrains it on the MIMIC-III corpus, demonstrating the importance of leveraging temporal information from sequences of notes on the task of prolonged mechanical ventilation prediction. Lastly, Lewis et al. [102] conduct an in-depth exploration of the state-of-the-art in biomedical and clinical pretrained language models as of late 2020, benchmarking a variety of models on the most extensive set of benchmark tasks compiled thus far. Notably, they perform an ablation study along three dimensions of interest—vocabulary, corpora, and model size—by training different versions of RoBERTa [111] models from scratch, with results suggesting that: learning a domain-specific vocabulary instead of using the default vocabulary from the general domain can be beneficial; the specificity of the training corpora does affect performance on downstream tasks; and large model size correlates with generally higher downstream performance.

Thanks to the easy-to-use design principle of Huggingface's Transformers library, using these pretrained models out of the box has become as easy as writing a few lines of python code, and domain-specific pretrained language models have become a standard part of the clinical NLP toolkit for practitioners.

Despite the rapid progress, there are several aspects in this line of research that must be further studied. First, as touched upon by Lewis et al., the idea of learning a clinical language-specific vocabulary instead of using the default vocabulary from pretrained models trained on general English corpora has been the subject of many clinical NLP researchers' curiosity. It is intuitively obvious that, given the vocabulary is usually constructed based on some simple language model trained on the corpus (typically based on unigram or byte-pair encoding), the resulting vocabulary would reflect the particular structure of the sublanguage that underlies the corpus. Whether the distribution of tokens in the learned vocabulary adheres closely enough to the sublanguage of interest (in our case the clinical language) is an empirical question. One of the main obstacles to conducting an in-depth study of clinical vocabularies in the era of transformer-based language models is the computational resources required to run such a study. Such studies would not be able to leverage the benefits of transfer learning by simply using general domain language models like BERT-base as is typically done because all of the weights in the pretrained models were trained jointly with the vocabulary embeddings specific to the corpora on which the models were trained; training with a specialized vocabulary would require starting from scratch. Moreover, the validation of such specialized vocabularies would involve training many variants of language models from scratch and evaluating them on a set of benchmark tasks. Since the current trend of further pretraining models from general NLP with non-specialized vocabularies has been sufficiently effective, the question of training such specialized vocabularies has not been prioritized.

Second, all existing clinical pretrained language models inherit the issue of limited context window sizes from the set of high-performing language models optimized for sentence-level representations (e.g. BERT, XLNet, RoBERTa, and Electra), typically at 512 tokens. This presents immediate problems for clinical notes, which are often longer than 512 tokens. We're forced to either truncate documents past the maximum sequence length of 512 tokens or to manually break down the documents into multiple chunks of approximately 512 tokens and then aggregate their predictions either as a post-processing step or with an additional learned module on top of the model. In any case, there is a nontrivial amount of information loss (from discarding parts of the notes or from breaking down long-range information through chunking), and current inelegant solutions end up introducing extra avenues of uncertainty and arbitrary decisions in pre/post-processing steps. While attempts to mitigate the problem of limited context window size and scalability in transformer-based language models [152] have yielded more efficient models that can handle longer sequences, this line of research has not yet been adapted into the clinical domain.

Lastly, the limited array of clinical text corpora is an important obstacle that needs to be addressed. MIMIC-III is constructed from a database of ICU visits, which are not representative of the entirety of medical practice. The fact that most clinical language models are trained on the MIMIC-III corpus introduces substantial biases that limit the generalizability and usefulness of findings that are based on those models. While the lack of clinical text corpora is one of the most difficult problems to address due to institutional and legal constraints, recent advances in de-identification methods, the growing acceptance of the research value of clinical text, and efforts to facilitate federated learning and interoperability in the medical community have made it more feasible to have alternative clinical text corpora, even if access is still limited to authorized individuals. While these models trained on custom non-MIMIC corpora might not be shared publicly, insights generated from their training and application can still be shared.

With the undeniable prevalence of these pretrained language models in clinical NLP, it is becoming increasingly important to address these questions in order for the field to continue progressing.

There is also a new but rapidly growing intersection between NLP and GRL. So far, this intersection is dominated by methods that attempt to incorporate knowledge graphs into the NLP pipeline for knowledge-intensive tasks such as question answering [177], natural language generation [38], conversational AI [30], and information extraction [96]. Galkin provides informative overviews of papers related to knowledge graphs and NLP in recent major AI conferences[8], showcasing the impressive growth of the intersection in just the past two years.

The combined application of recent NLP and GRL techniques has also been emerging in the clinical domain. Most instances of applications of novel NLP or GRL techniques on clinical data so far involve either only clinical text or only structured clinical information such as medical codes, and on the rare occasions of multimodal learning (specifically a setting in which clinical text and medical codes along with other clinical information are combined), most of them involve simply concatenating the representations of the separate modalities to obtain a combined representation. While the simple concatenation followed by additional modules can be seen as simplistic and inelegant, its prevalence is an indication of the difficulty of truly doing relational and multimodal learning in an effective way. Some specific examples of recent attempts to apply novel NLP and/or GRL techniques on clinical data are summarized next.

Choi et al. [44] propose Graph Convolutional Transformers, a modified version of stacked transformer models that jointly learns the latent structure of medical codes in EHR visits. Precomputed co-occurrence statistics between diagnosis and procedure codes as well as hand-engineered attention masks that selectively mask specified relations are used to guide the learning process. The model is trained on both the eICU collaborative research dataset [128] and synthetic encounter records for several medical prediction tasks. While GCT offers an interesting perspective on latent structure learning using the self-attention mechanism, it has several limitations. It includes only two or three modalities (diagnosis, procedures, lab tests) and no text, it is too engineering-heavy and difficult to extend to other settings. For instance, constructing the prior guide matrices and the attention masks would require untenable hand-engineering efforts in order to handle more modalities, let alone text.

[8] https://migalkin.github.io/

Steinberg et al. [147] argue that language models are an effective way to learn the representation of EHR data and use de-identified EHR data from Stanford Hospital and Lucile Packard Children's Hospital amounting to 3.4 million patient records spanning 1990 through 2018. While they use diagnosis, procedures, medication, laboratory test orders in the form of their respective codes (ICD10, CPT or HCPCS, RXCUI, and LOINC), they do not use quantitative data or text. A simple Gated Recurrent Unit-based language model is trained on these codes to produce the patient embeddings, and a thorough comparison to existing patient representation learning methods is provided. However, there is a major missed opportunity in the lack of attempts for relational learning (between codes from the various terminologies) or the joint learning of text.

BEHRT [103] adopts the BERT model architecture and trains on data from 1.6 million individuals from the Clinical Practice Research Datalink in the UK containing longitudinal primary care data. Instead of text, it uses sequences consisting of diagnosis codes for all the visits pertaining to a patient in order to learn how to predict the likelihood of 301 conditions in future visits.

Lee et al. [100] take a graph-based approach to learning the representation of sequences of patients' medical records. They construct multi-modal graphs based on patient records, consisting of ICD-9 diagnosis codes and UMLS medical concepts extracted from clinical notes using Metamap [10]. The resulting heterogeneous graph of diagnosis codes, medical concepts, and patient visits are passed through a GCN and LSTM to obtain the embeddings.

MedGraph [78] represents visit-code associations in an attributed bipartite graph and the temporal sequence of visits through a Gaussian point process to produce Gaussian embeddings for visits and codes for several medical risk prediction tasks.

Rocheteau et al. [133] addresses the fact that most multimodal attempts use simple concatenation and provides an alternative approach that combines GNNs and

LSTMs. Specifically, they construct a patient network using a measure of patient similarity based on diagnosis codes, and they use LSTMs to encode temporal features (i.e. physiological time series) and GNNs to encode patient neighborhood information to predict mortality and length of stay on the eICU database.

None of these methods have directly attempted to combine text with structured data, likely due to the difficulty of such an endeavor. An important point that needs to be addressed when discussing GRL, a point that is most pertinent in the clinical domain, is that most GRL methods assume that a graph structure is given, and focus on ways to embed the given graph into a distributed representation amenable to machine learning. The open challenge is the learning or inferring of graphs or relational structure from data without explicit structure (like text, set of medical codes, images, measurements, etc). Latent graph learning or inference, a fundamental problem in GRL, is the bridge between unstructured/semi-structured data and existing GRL (or even NLP) methodologies. Latent graph learning also has the potential to actually improve upon existing graph structures. For example, learning an ontology that is better than existing ones or learning a patient graph that goes beyond querying and merging existing data components and their schema would be an incredible step forward in medical informatics research.

The subsequent chapters of this dissertation document my PhD journey through the intersection of NLP, GRL, and clinical data, starting with clinical NLP in the post-BERT era, followed by explorations in the graph learning space involving biomedical knowledge graphs and GNNs, and concluding with a novel integrative method that ties together the various topics of interest.

Chapter 3

Generating Contextual Embeddings for Emergency Department Chief Complaints using BERT

3.1 Background

Patient care in the emergency department (ED) is guided by the patient's chief complaint [69, 114, 115]. Collected during the first moments of the patient encounter, a chief complaint is a concise statement regarding the patient's medical history, current symptoms, and reason for visit. While a chief complaint can be represented in a structured format with predefined categories, it is often captured in unstructured, free-text descriptions of varying length and quality [72]. Moreover, even when chief complaints are stored in a structured format, there exists no standard nomenclature or guidance on how they should be categorized [85, 9]. As a consequence, administrators and researchers frequently find chief complaint data difficult to use for downstream tasks such as quality improvement initiatives and predictive analytics [48]. Thus, the secondary use of chief complaint data in daily operational decisions and research has been hampered by its form and representation.

Recent advances in natural language processing (NLP) provide an opportunity to address many of the challenges of chief complaint data. Contextual language models are able to generate dense vector representations, or embeddings, of free-text data such that semantically similar words or documents are mapped to nearby points in vector space [123, 55, 158]. Such methods have been successfully applied in the medical domain [45, 12, 13, 18, 189, 145, 148]. Recent work has used contextual language models to generate embeddings for chief complaints in the primary care setting [156].

Embeddings for ED chief complaints have several desirable properties. First, in addition to better semantic representation, the size of the embedding space can be chosen to meet a particular use case. For example, free-text chief complaints could be represented in 10, 50, or 200 dimensions depending on desired accuracy and computing resources. Second, a more dense, contextually-informed representation could enhance clustering techniques aimed at deriving a standardized ontology of ED chief complaints [48, 34, 35]. Lastly, a model that maps free-text data to such an ontology could be shared among healthcare institutions and research entities to minimize the variability in how chief complaint labels are assigned from ED to ED [85, 89, 68].

In this study, we expand on prior work by applying Bidirectional Encoder Representations from Transformers (BERT), a state-of-the-art NLP model, on a dataset of 1.8 million free-text ED chief complaints from a healthcare system covering seven independent EDs [55**?**]. We show that the contextual embeddings generated by BERT accurately predict provider-assigned chief complaint labels and map semantically similar chief complaints to nearby points in vector space.

Table 3.1: Exclusion thresholds for chief complaint label frequency

Cutoff Threshold	Minimum Count	Dataset Size	Unique Labels
0.01%	188	1,859,599	434
0.02%	376	1,837,277	350
0.04%	752	1,786,604	260
0.08%	1,504	1,709,206	188
0.16%	3,008	1,562,565	117
0.32%	6,016	1,376,629	73
0.64%	12,032	1,001,417	29

3.2 Materials and Methods

Retrospective data on all adult and pediatric emergency department (ED) visits was obtained from a large healthcare system covering the period of March 2013 to July 2019, with a combined annual census of approximately 500,000 visits across seven independent EDs, three of which are community hospital-based. The centralized data warehouse for the healthcare system (Epic, Verona, WI) was queried for chief complaint data. This study was approved, and the informed consent process waived, by the Human Investigation Committee at the authors' institution (HIC 2000025236). Given the skewed distribution of chief complaint labels, where the 25 most common labels out of a total of 1145 account for roughly half of the dataset, chief complaint labels that comprised less than 0.01%, or 1 in 10,000, of all visits were excluded. The cut-off threshold was then incremented on a log scale to create seven datasets of decreasing sparsity, as shown in Table 3.1

3.3 Model Training

For each of the seven datasets, all samples were randomly split into training (90%) and test (10%) sets. The classification task was to predict the provider-assigned label from the free-text chief complaint. Given the clinical nature of our dataset, we used a version of clinical BERT pre-trained on the MIMIC corpus [90]. Using the open source library PyTorch, we trained each model for three epochs on three GTX 1080 Ti GPUs. Each epoch on the full dataset took about an hour using `per_gpu_train_batch_size` of 144 and `per_gpu_eval_batch_size` of 2400. Hyperparameter tuning beyond the default values for BERT fine-tuning did not yield noticeable gains in performance, with the test accuracies converging to the same range of values for any reasonable configuration. A learning rate of 1e-4 and `max_seq_length` of 64 was used. The implementation code is on github available[1]. Notably, the repository also includes an easy-to-use script with instructions to generate predictions for custom chief complaint datasets.

3.4 Error Analysis

Having hundreds of potential labels with considerable semantic overlap (e.g. FACIAL LACERATION, LACERATION, HEAD LACERATION, FALL, FALL>65) justifies taking into account the top few predictions rather than just the top 1. We hypothesized that the redundancy and noise in the label space would be responsible for the majority of the model's errors and a priori determined to examine a random sample of errors, as well as look at the most frequent kinds of mislabeling for common chief complaint labels.

3.5 Embedding Visualization

The embedding for each free-text chief complaint was extracted as the final 768-dimensional layer of the BERT classifier. We took the mean of the embeddings across each chief complaint label and visualized the averaged, label-specific embeddings in a 2-dimensional space using t-SNE [157]. More specifically, the mean of the 768-dimensional embeddings across each chief complaint label was reduced to

[1] https://github.com/dchang56/chief_complaints

two dimensions using the Rtsne package (v. 0.15) in R with the following default hyperparameters: `initial_dims` = 50, `perplexity` = 30, `theta` = 0.5. To enhance readability of the figure, we limited the number of visualized labels to 188 by using a cutoff threshold of 0.08%. The ggrepel and ggplot2 packages in R were used for plot generation. Clusters were determined via Gaussian mixture modeling with the optimal number selected by silhouette analysis [135].

3.6 Results

In the defined query time period, there were an initial 2,154,862 visits among 736,570 patients. 355,497 (16.4%) visits from 65,737 (8.9%) patients were removed for absence of either a structured or unstructured chief complaint. Among chief complaint labels, 43 of the 1,145 labels were removed because of the absence of any visit with unstructured text. An additional 668 labels were removed after filtering out labels that comprised less than 0.01%, or 1 in 10,000, of all visits.

The models achieved increasing performance with higher label-frequency cutoff thresholds. All models passed 90% Top-4 accuracy on the test sets, as shown in Figure 3.1. Common types of mislabeling for the frequent chief complaint labels, as well as labels with the lowest accuracies, are shown in Figure 3.2. The interquartile range for Top-5 accuracies amongst the chief complaint labels was 74.0% – 92.3%.

Manual error analysis showed that many errors were due to the problem of redundancy and noise in the label space. In some cases, the predictions of the model were more suitable than the provider-assigned labels. We show ten representative examples in Table 3.2.

Figure 3.3 shows the t-SNE visualization of averaged embeddings for common chief complaint labels, clustered via Gaussian mixture modeling. Using the silhouette analysis, 15 was chosen to be the optimal number of clusters. A cutoff-threshold of

Table 3.2: Examples of Chief Complaints and Their Corresponding Top-k Predictions

	Chief Complaint	**Top-k Predictions**
Correctly classified at second prediction	"right third finger injured in door"	FINGER INJURY, **HAND PAIN**
	"pt comes to er with cc peice of plastic stuck to back of left ear from earing"	FOREIGN BODY IN EAR, **EAR PROBLEM**
	"vomiting for days, increasing yesterday. pos home preg test on Saturday"	EMESIS, **EMESIS DURING PREGNANCY**
	"both eyes swollen & itchy & tearing after his nap"	EYE SWELLING, **EYE PROBLEM**
	"fall at 0300 today, rt side weakness"	FALL, **FALL>65**
Correctly classified at fifth prediction	"Felt like heart was pounding history of cabg. missed metoprolol for about 3 days."	PALPITATIONS, RAPID HEART RATE, TACHYCARDIA, IRREGULAR HEART BEAT, **CHEST PAIN**
	"2 weeks of sore throat, aches, dry cough. Denies intervention."	SORE THROAT, COLD LIKE SYMPTOMS, URI, COUGH, **FLU LIKE SYMPTOMS**
	"fall down 5 stairs lace to right eyebrow"	FALL, FACIAL LACERATION, LACERATION, FALL>65, **HEAD LACERATION**
	"fever to 101, diarrhea, vomiting"	FEVER-9 WEEKS TO 74 YEARS, FEVER, EMESIS, ABDOMINAL PAIN, **FEVER-8 WEEKS OR LESS**
	"blister on back of foot."	BLISTER, FOOT PAIN, FOOT INJURY, FOOT SWELLING, **SKIN PROBLEM**

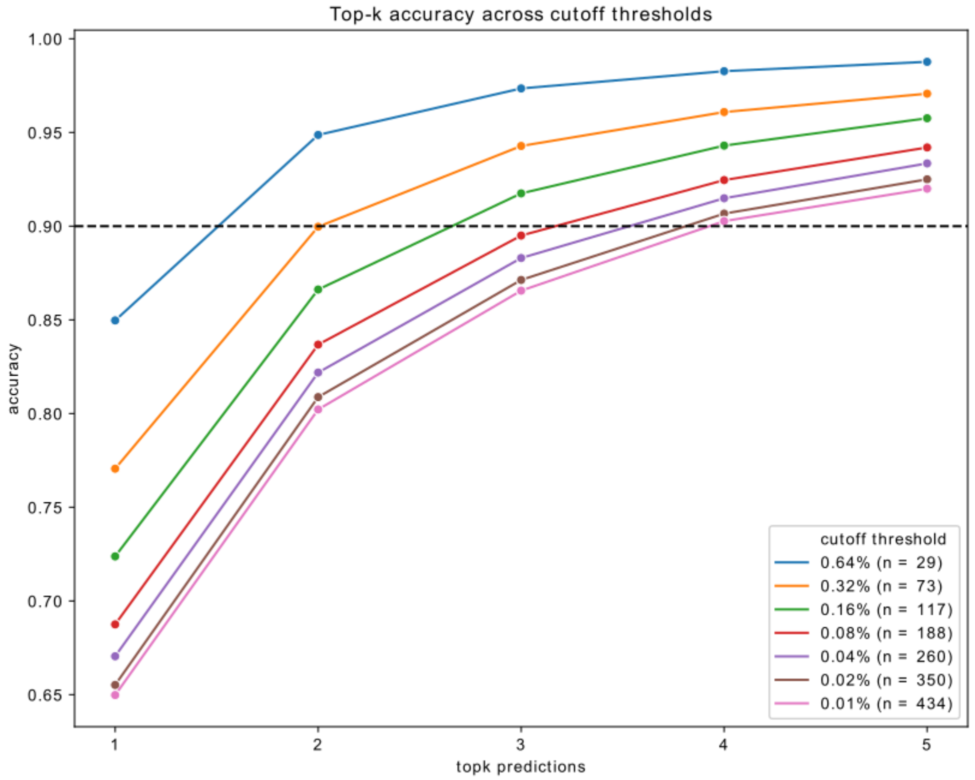


Figure 3.1: Model Performance for Top-1 to Top-5 Accuracy. Label-frequency cutoff thresholds are represented by colors. The accuracy increases drastically when taking into account the first few predictions. Dotted line shows 90% accuracy.

0.08% (188 chief complaint labels) was used for readability in a 2-dimensional space.

3.7 Discussion

By applying BERT on a dataset of 1.8 million ED chief complaints from a healthcare system covering seven independent EDs, we derive embeddings for chief complaints that accurately predict provider-assigned labels as well as map semantically similar chief complaints to nearby points in vector space. These embeddings, 768 dimensions in their original form, are significantly more practical for downstream tasks compared to free-text or categorical data. Our aim is not to prove the superiority of BERT over preexisting NLP methods but to demonstrate how BERT can be successfully applied to clinical data in emergency care.

There has been previous work deriving embeddings for medical concepts, patient-to-provider messages, and primary care chief complaints [45, 148, 156], but our study is the first to derive embeddings for ED chief complaints. We present our embeddings explicitly rather than as a hidden layer in a prediction task as these embeddings may be instrumental in multiple downstream tasks, such as calculating similarity measures between chief complaints to determine whether ED bounce-backs are due to a related cause [132] or creating a data-driven ontology of chief complaints without a need for expert-panel opinion [89, 85]. Future work will focus on creating such an ontology through in-depth cluster analysis and exploring whether the embeddings contain clinical information regarding a patient's acuity, such as the likelihood of hospital admission or 30-day mortality.

Our study has several limitations. One limitation is the noise inherent in the default set of chief complaint labels provided by our electronic health record system. Of the 1145 default categories, 153 have one or no instance out of 1.87 million visits, while 472 account for 99% of the visits. Labels such as "OTHER" and "MEDICAL" provide little to no information in an emergency care setting. Some labels are synonyms (e.g. "dyspnea" and "shortness of breath"; "otalgia" and "ear pain"), while many more are hypo/hypernyms of one another (e.g. "fall" and "fall>65"; "migraine" and "known dx migraine"). Such issues highlight the need to develop a principled and data-driven ontology for ED chief complaints. Despite the noise in the data, the model was able to learn a rich representation of chief complaints and generate reasonable predictions of their labels. In fact, many of the predictions that resulted in errors were more suitable than the ground truth labels, suggesting that the model did not overfit to the data. Another limitation of the study is that free-text chief complaints often list several comorbid signs and symptoms, making it difficult to choose a single ground truth label. This raises concerns about whether the prediction task should be set up as a multi-label classification task. Finally, our model was trained only on free-text data, without any other patient information. Including non-textual patient data such as demographics, vital signs, and hospital usage statistics may improve performance, as shown in many prediction tasks [84, 83]. Further studies will be needed to assess the validity of this approach.

3.8 Conclusion

The BERT model was able to learn a rich representation of chief complaints and generate reasonable predictions of their labels despite the inherent noise in the label space. The learned embeddings accurately predicted provider-assigned chief complaint labels and mapped semantically similar chief complaints to nearby points in vector space. Such a model may be used to automatically map free-text chief complaints to structured fields and to derive a standardized, data-driven ontology of chief complaints for healthcare institutions.

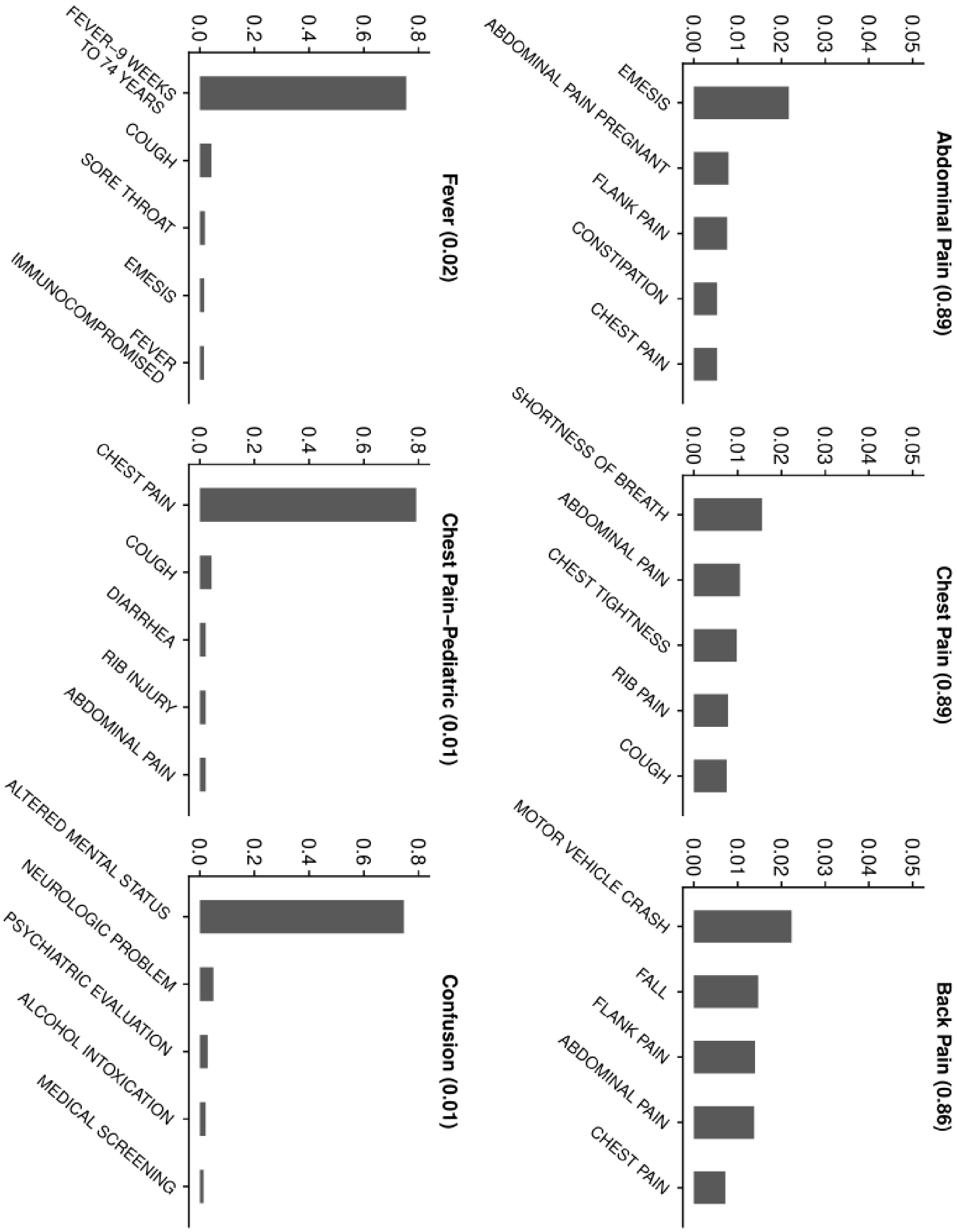


Figure 3.2: Common Types of Mislabeling for Select Chief Complaint Labels. Top row shows three of the most common chief complaint labels, with their accuracies shown within parentheses. Bottom row shows three chief complaint labels with lowest accuracies. Y-axis shows frequency of error. Note that even for low performing chief complaint labels, a high percentage of errors are due to semantic overlap.

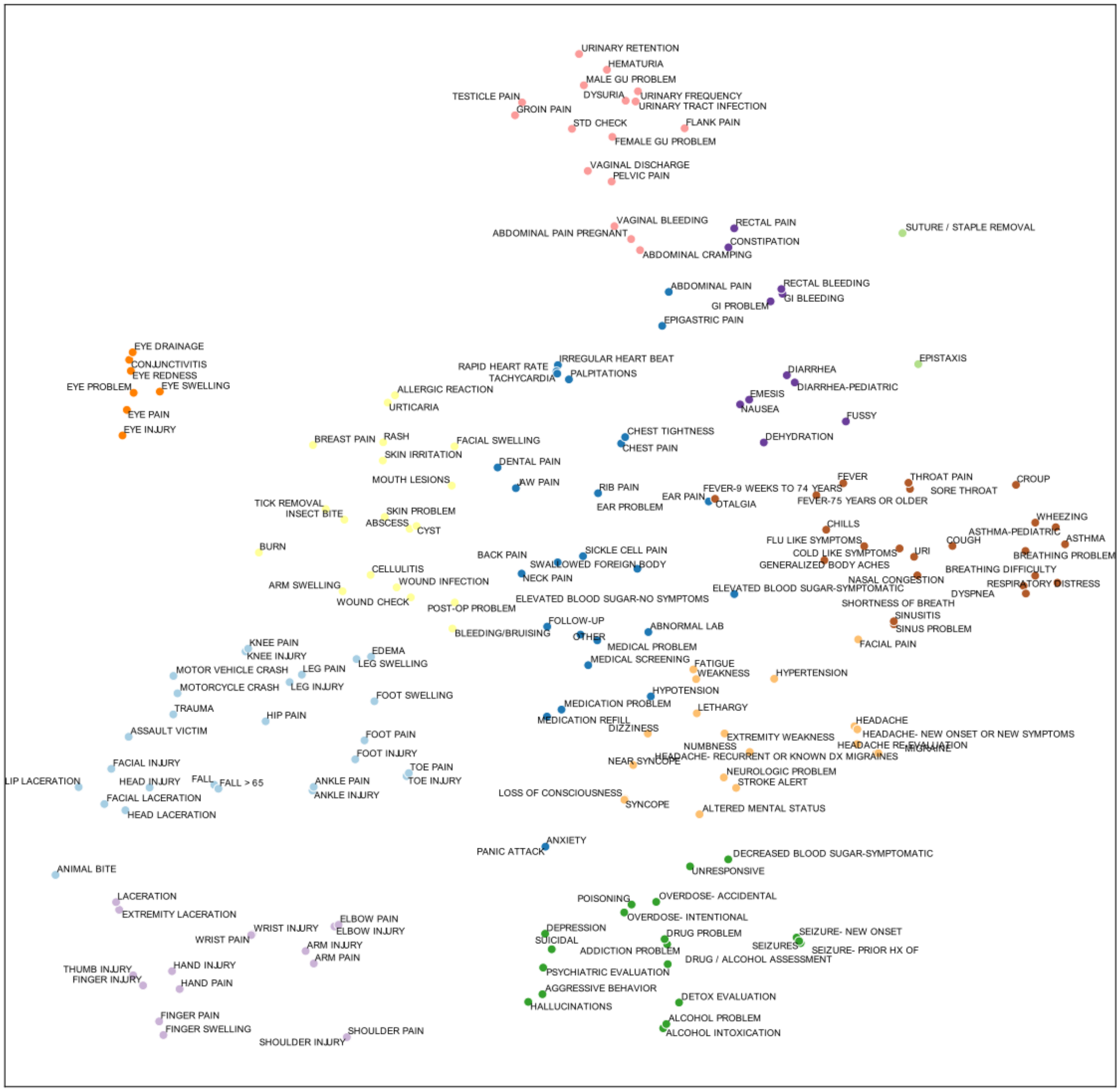


Figure 3.3: t-SNE Visualization of Averaged Embeddings of Common Chief Complaint Labels. The embeddings are distributed in a clinically meaningful way, with related concepts embedded close to each other and broader types of chief complaints clustered together. Note that t-SNE is a stochastic algorithm and, while it preserves local structure of the data, does not completely preserve its global structure. The text labels have been jittered to enhance readability. Colored groupings represent clusters as determined by gaussian mixture modeling.

Chapter 4

Benchmark and Best Practices for Biomedical Knowledge Graph Embeddings

4.1 Abstract

Much of biomedical and healthcare data is encoded in discrete, symbolic form such as text and medical codes. There is a wealth of expert-curated biomedical domain knowledge stored in knowledge bases and ontologies, but the lack of reliable methods for learning knowledge representation has limited their usefulness in machine learning applications. While text-based representation learning has significantly improved in recent years through advances in natural language processing, attempts to learn biomedical concept embeddings so far have been lacking. A recent family of models called knowledge graph embeddings have shown promising results on general domain knowledge graphs, and we explore their capabilities in the biomedical domain. We train several state-of-the-art knowledge graph embedding models on the SNOMED-CT knowledge graph, provide a benchmark with comparison to existing methods and

in-depth discussion on best practices, and make a case for the importance of leveraging the multi-relational nature of knowledge graphs for learning biomedical knowledge representation. The embeddings, code, and materials will be made available to the community[1].

4.2 Introduction

A vast amount of biomedical domain knowledge is stored in knowledge bases and ontologies. For example, SNOMED Clinical Terms (SNOMED-CT)[2] is the most widely used clinical terminology in the world for documentation and reporting in healthcare, containing hundreds of thousands of medical terms and their relations, organized in a polyhierarchical structure. SNOMED-CT can be thought of as a knowledge graph: a collection of triples consisting of a head entity, a relation, and a tail entity, denoted (h, r, t). SNOMED-CT is one of over a hundred terminologies under the Unified Medical Language System (UMLS) [22], which provides a metathesaurus that combines millions of biomedical concepts and relations under a common ontological framework. The unique identifiers assigned to the concepts as well as the Resource Release Format (RRF) standard enable interoperability and reliable access to information. The UMLS and the terminologies it encompasses are a crucial resource for biomedical and healthcare research.

One of the main obstacles in clinical and biomedical natural language processing (NLP) is the ability to effectively represent and incorporate domain knowledge. A wide range of downstream applications such as entity linking, summarization, patient-level modeling, and knowledge-grounded language models could all benefit from improvements in our ability to represent domain knowledge. While recent advances in NLP have dramatically improved textual representation [8], attempts to learn

[1] https://github.com/dchang56/snomed_kge

[2] https://www.nlm.nih.gov/healthit/snomedct

analogous dense vector representations for biomedical concepts in a terminology or knowledge graph (*concept embeddings*) so far have several drawbacks that limit their usability and wide-spread adoption. Further, there is currently no established best practice or benchmark for training and comparing such embeddings. In this paper, we explore knowledge graph embedding (KGE) models as alternatives to existing methods and make the following contributions:

- We train five recent KGE models on SNOMED-CT and demonstrate their advantages over previous methods, making a case for the importance of leveraging the multi-relational nature of knowledge graphs for biomedical knowledge representation.
- We establish a suite of benchmark tasks to enable fair comparison across methods and include much-needed discussion on best practices for working with biomedical knowledge graphs.
- We also serve the general KGE community by providing benchmarks on a new dataset with real-world relevance.
- We make the embeddings, code, and other materials publicly available and outline several avenues of future work to facilitate progress in the field.

4.3 Related Work and Background

4.3.1 Biomedical concept embeddings

Early attempts to learn biomedical concept embeddings have applied variants of the skip-gram model [113] on large biomedical or clinical corpora. Med2Vec [41] learned embeddings for 27k ICD-9 codes by incorporating temporal and co-occurrence information from patient visits. Cui2Vec [17] used an extremely large collection of multimodal medical data to train embeddings for nearly 109k concepts under the UMLS.

These corpus-based methods have several drawbacks. First, the corpora are inaccessible due to data use agreements, rendering them irreproducible. Second, these methods tend to be data-hungry and extremely data inefficient for capturing domain knowledge. In fact, one of the main limitations of language models in general is their reliance on the distributional hypothesis, essentially making use of mostly co-occurrence level information in the training corpus [126]. Third, they do a poor job of achieving sufficient concept coverage: Cui2Vec, despite its enormous training data, was only able to capture 109k concepts out of over 3 million concepts in the UMLS, drastically limiting its downstream usability.

A more recent trend has been to apply network embedding (NE) methods directly on a knowledge graph that represents structured domain knowledge. NE methods such as Node2Vec [70] learn embeddings for nodes in a network (graph) by applying a variant of the skip-gram model on samples generated using random walks, and they have shown impressive results on node classification and link prediction tasks on a wide range of network datasets. In the biomedical domain, CANode2Vec [97] applied several NE methods on single-relation subsets of the SNOMED-CT graph, but the lack of comparison to existing methods and the disregard for the heterogeneous structure of the knowledge graph substantially limit its significance.

Notably, Snomed2Vec [2] applied NE methods on a clinically relevant multi-relational subset of the SNOMED-CT graph and provided comparisons to previous methods to demonstrate that applying NE methods directly on the graph is more data efficient, yields better embeddings, and gives explicit control over the subset of concepts to train on. However, one major limitation of NE approaches is that they relegate relationships to mere indicators of connectivity, discarding the semantically rich information encoded in multi-relational, heterogeneous knowledge graphs.

We posit that applying KGE methods on a knowledge graph is more principled and should therefore yield better results. We now provide a brief overview of the KGE literature and describe our experiments in Section 4.3.2.

4.3.2 Knowledge Graph Embeddings

Knowledge graphs are collections of facts in the form of ordered triples ($\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$), where entity $\mathbf{h}$ is related to entity $\mathbf{t}$ by relation $\mathbf{r}$. Because knowledge graphs are often incomplete, an ability to infer unknown facts is a fundamental task (link prediction). A series of recent KGE models approach link prediction by learning embeddings of entities and relations based on a scoring function that predicts a probability that a given triple is a fact.

RESCAL [119] represents relations as a bilinear product between subject and object entity vectors. Although a very expressive model, RESCAL is prone to overfitting due to the large number of parameters in the full rank relation matrix, increasing quadratically with the number of relations in the graph.

DistMult [178] is a special case of RESCAL with a diagonal matrix per relation, reducing overfitting. However, by limiting linear transformations on entity embeddings to a stretch, DistMult cannot model asymmetric relations.

ComplEx [154] extends DistMult to the complex domain, enabling it to model asymmetric relations by introducing complex conjugate operations into the scoring function.

SimplE [91] modifies Canonical Polyadic (CP) decomposition [80] to allow two embeddings for each entity (head and tail) to be learned dependently.

A recent model TuckER [14] is shown to be a fully expressive, linear model that subsumes several tensor factorization based approaches including all models described above.

TransE [24] is an example of an alternative *translational* family of KGE models, which regard a relation as a translation (vector offset) from the subject to the object entity vectors. Translational models have an additive component in the scoring function, in contrast to the multiplicative scoring functions of bilinear models.

RotatE [150] extends the notion of translation to rotation in the complex plane, enabling the modeling of symmetry/antisymmetry, inversion, and composition patterns in knowledge graph relations.

We restrict our experiments to five models due to their available implementation under a common, scalable platform [191]: TransE, ComplEx, DistMult, SimplE, and RotatE.

4.4 Experimental Setup

4.4.1 Data

Given the complexity of the UMLS, we detail our preprocessing steps to generate the final dataset. We subset the 2019AB version of the UMLS to `SNOMED_CT_US` terminology, taking all active concepts and relations in the MRCONSO.RRF and MRREL.RRF files. We extract semantic type information from MRSTY.RRF and semantic group information from the Semantic Network website[3] to filter concepts and relations to 8 broad semantic groups of interest: Anatomy (**ANAT**), Chemicals & Drugs (**CHEM**), Concepts & Ideas (**CONC**), Devices (**DEVI**), Disorders (**DISO**), Phenomena (**PHEN**), Physiology (**PHYS**), and Procedures (**PROC**). We also exclude specific semantic types deemed unnecessary.

The resulting list of triples comprises our final knowledge graph dataset. Note that the UMLS includes reciprocal relations (`ISA` and `INVERSE_ISA`), making the graph bidirectional. A random split results in train-to-test leakage, which can inflate the performance of weaker models [54]. We fix this by ensuring reciprocal relations are in

[3] https://semanticnetwork.nlm.nih.gov

the same split, not across splits. Descriptive statistics of the final dataset are shown in Table 4.1. After splitting, we also ensure there are no unseen entities or relations in the validation and test sets by simply moving them to the train set.

Descriptions	**Statistics**
Entities	293,884
Relation types	170
Facts	2,073,848
- Train	1,965,032
- Valid / Test	48,936 / 49,788

Table 4.1: Statistics of the final SNOMED dataset.

4.4.2 Implementation

Considering the non-trivial size of SNOMED-CT and the importance of scalability and consistent implementation for running experiments, we use GraphVite [191] for the KGE models. GraphVite is a graph embedding framework that emphasizes scalability, and its speedup relative to existing implementations is well-documented[4]. While the backend is written largely in C++, a Python interface allows customization. We make our customized Python code available. We use the five models available in GraphVite in our experiments: TransE, ComplEx, DistMult, SimplE, and RotatE. While we restrict our current work to these models, future work should also consider other state-of-the-art models such as TuckER [14] and MuRP [15], especially since MuRP is shown to be particularly effective for graphs with hierarchical structure. Pretrained embeddings for Cui2Vec and Snomed2Vec were used as provided by the authors, with dimensionality 500 and 200, respectively.

All experiments were run on 3 GTX-1080ti GPUs, and final runs took ∼6 hours on a single GPU. Hyperparameters were either tuned on the validation set for each model: `margin` (4, 6, 8, 10) and `learning_rate` (5e-4, 1e-4, 5e-5, 1e-5); set: `num_negative`

[4] https://github.com/DeepGraphLearning/graphVite

(60), `dim` (512), `num_epoch` (2000); or took default values from GraphVite.

4.4.3 Evaluation and Benchmark

KGE Link Prediction

A standard evaluation task in the KGE literature is link prediction. However, NE methods also use link prediction as a standard evaluation task. While both predict whether two nodes are connected, NE link prediction performs binary classification on a balanced set of positive and negative edges based on the assumption that the graph is complete. In contrast, knowledge graphs are typically assumed incomplete, making link prediction for KGE a ranking-based task in which the model's scoring function is used to rank candidate samples without relying on ground truth negatives. In this paper, link prediction refers to the latter ranking-based KGE method.

Candidate samples are generated for each triple in the test set using all possible entities as the target entity, where the target can be set to `head`, `tail`, or `both`. For example, if the target is `tail`, the model predicts scores for all possible candidates for the tail entity in (h, r, ?). For a test set with 50k triples and 300k possible unique entities, the model calculates scores for fifteen billion candidate triples. The candidates are filtered to exclude triples seen in the train, validation, and test sets, so that known triples do not affect the ranking and cause false negatives. Several ranking-based metrics are computed based on the sorted scores. Note that SNOMED-CT contains a *transitive closure* file, which lists explicit transitive closures for the hierarchical relations `ISA` and `INVERSE_ISA` (if A `ISA` B, and B `ISA` C, then the transitive closure includes A `ISA` C). This file should be included in the file list used to filter candidates to best enable the model to learn hierarchical structure.

Typical link prediction metrics include Mean Rank (MR), Mean Reciprocal Rank (**MRR**), and Hits@k (**H@k**). MR is considered to be sensitive to outliers and unreliable as a metric. Gu et al. [71] proposed using Mean Quantile (**MQ**) as a more robust alternative to MR and MRR. We use MQ_{100} as a more challenging version of MQ that introduces a cut-off at the top 100th ranking, appropriate for the large numbers of possible entities. Link prediction results are reported in Table 4.2.

Embedding Evaluation

For fair comparison with existing methods, we perform some of the benchmark tasks for assessing medical concept embeddings proposed by Beam et al. However, we discuss their methodological flaws in Section 4.7 and suggest more appropriate evaluation methods.

Since non-KGE methods are not directly comparable on tasks that require both relation and concept embeddings, to compare embeddings across methods we perform entity semantic classification, which requires only concept embeddings.

We generate a dataset for entity classification by taking the intersection of the concepts covered in all (7) models, comprising 39k concepts with 32 unique semantic types and 4 semantic groups. We split the data into train and test sets with 9:1 ratio, and train a simple linear layer with 0.1 dropout and no further hyperparameter tuning. The single linear layer for classification assesses the linear separability of semantic information in the entity embedding space for each model. Results for semantic type and group classification are reported in Table 4.3.

4.5 Visualization

We first discuss the embedding visualizations obtained through LargeVis [151], an efficient large-scale dimensionality reduction technique available as an application in GraphVite.

Figure 4.1 shows concept embeddings for RotatE, ComplEx, Snomed2Vec, and Cui2Vec, with colors corresponding to broad semantic groups. Cui2Vec embeddings

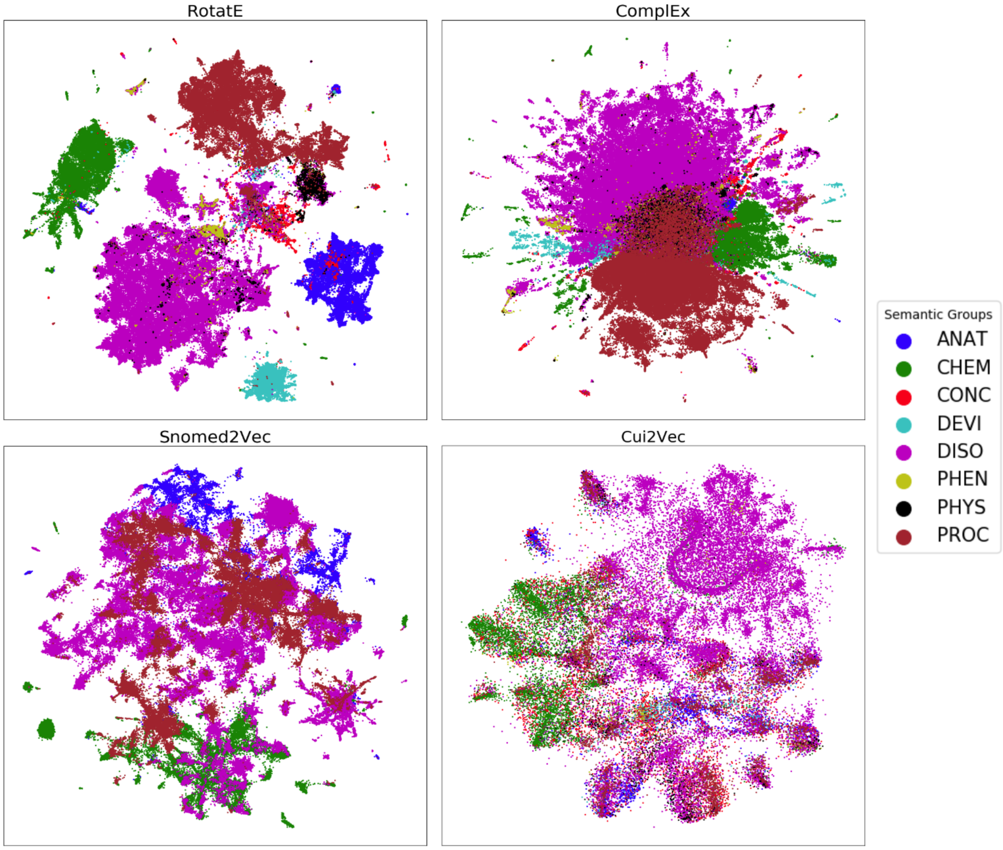


Figure 4.1: Concept embedding visualization (RotatE, ComplEx, Snomed2Vec, Cui2Vec) by semantic group.

show structure but not coherent semantic clusters. Snomed2Vec shows tighter groupings of entities, though the clusters are patchy and scattered across the embedding space. ComplEx produces globular clusters centered around the origin, with clearer boundaries between groups. RotatE gives visibly distinct clusters with clear group separation that appear intuitive: entities of the Physiology semantic group (black) overlap heavily with those of Disorders (magenta); also entities under the Concepts semantic group (red) are relatively scattered, perhaps due to their abstract nature, compared to more concrete entities like Devices (cyan), Anatomy (blue), and Chemicals (green), which form tighter clusters.

Interestingly, the embedding visualizations for the 5 KGE models fall into 2 types: RotatE and TransE produce well-separated clusters while ComplEx, DistMult and SimplE produce globular clusters around the origin. Since the plots for each type

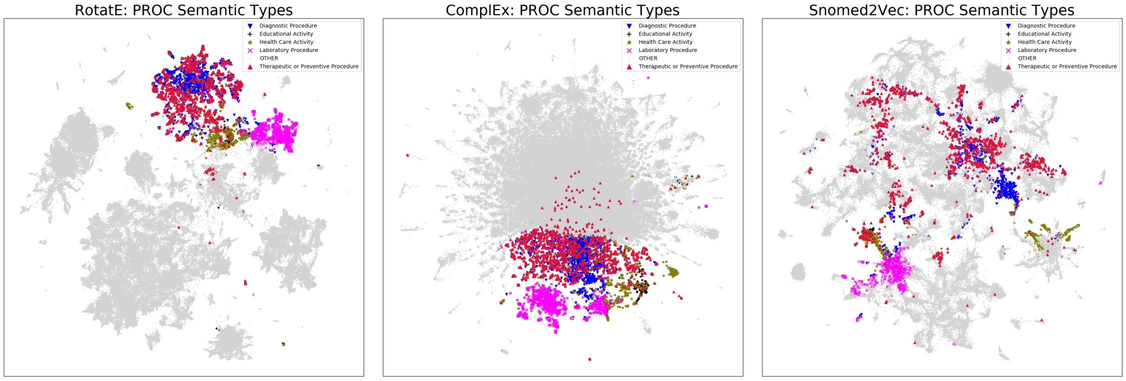


Figure 4.2: Visualization of selected semantic types under the Procedures semantic group for RotatE, ComplEx, and Snomed2Vec. Semantic types with more than 2,000 entities were subsampled to 1,200 for visibility. Cui2Vec (not shown) was similar to Snomed2Vec but more dispersed.

appear almost indistinguishable we show one from each (RotatE and ComplEx). We attribute the characteristic difference between the two model types to the nature of their scoring functions: RotatE and TransE have an additive component while ComplEx, DistMult and SimplE are multiplicative.

Figure 4.2 shows more fine-grained semantic structure by coloring 5 selected semantic types under the Procedures semantic group and greying out the rest. We see that RotatE produces subclusters that are also intuitive. *Laboratory procedures* are well-separated on their own, *health care activity* and *educational activity* overlap significantly, and *diagnostic procedures* and *therapeutic or preventative procedures* overlap significantly. ComplEx also reveals subclusters with globular shape, and Snomed2Vec captures *laboratory procedures* well but leaves other types scattered. These observations are consistent across other semantic groups.

While semantic class information is not the only significant aspect of SNOMED-CT, since the SNOMED-CT graph is largely organized around semantic group and type information, it is promising that embeddings learned (without supervision) preserve it.

4.6 Results

4.6.1 Link Prediction

Model	**MRR**	**MQ$_{100}$**	**H@1**	**H@10**
TransE	.346	.739	.212	.597
ComplEx	.461	.761	.360	.652
DistMult	.420	.752	.309	.626
SimplE	.432	.735	.337	.615
RotatE	.317	.742	.162	.599
TransE$_{FB}$	.294	-	-	.465
TransE$_{WN}$	.226	-	-	.501
RotatE$_{FB}$	.338	-	.241	.533
RotatE$_{WN}$	.476	-	.428	.571

Table 4.2: Link prediction results: for the 5 KGE models on SNOMED-CT (top); and for TransE and RotatE on two standard KGE datasets [150] (bottom).

Table 4.2 shows results for the link prediction task for the 5 KGE models on SNOMED-CT. Having no previous results to compare to, we include performance of TransE and RotatE on two standard KGE benchmark datasets for reference: FB15k-237 (14,541 entities, 237 relations, and 310,116 triples) and WN18RR (40,943 entities, 11 relations, and 93,003 triples). Given that SNOMED-CT is larger and arguably a more complex knowledge graph than the two datasets, the link prediction results suggest that the KGE models learn a reasonable representation of SNOMED-CT. We include sample model outputs for the top 10 entity scores for link prediction in the Supplements.

4.6.2 Embedding Evaluation and Relation Prediction

Test set accuracy for entity semantic type (**STY**) and semantic group (**SG**) classification are reported in Table 4.3. In accordance with the visualizations of semantic clusters (Figures 4.1 and 4.2), the KGE and NE methods perform significantly better than the corpus-based method (Cui2Vec). Notably, TransE and RotatE attain near-

	Entity Classification		Cosine-Sim Bootstrap			Relation Prediction		
Model	**SG (4)**	**STY (32)**	**ST**	**CA**	**Co**	**MRR**	**H@1**	**H@10**
Snomed2Vec	.944	.769	.387	**.903**	.894	-	-	-
Cui2Vec	.891	.673	.416	.584	.559	-	-	-
TransE	**.993**	**.827**	**.579**	.765	**.978**	.800	.727	**.965**
ComplEx	.956	.786	.249	.001	.921	.731	.606	.914
DistMult	.971	.794	.275	.014	.971	.734	.569	.946
SimplE	.953	.768	.242	.011	.791	**.854**	**.803**	.946
RotatE	**.995**	**.829**	.544	.242	.943	**.849**	**.799**	.957

Table 4.3: Results for (i) entity classification of semantic type and group (test accuracy); (ii) selected tasks from [17]; and (iii) relation prediction. Best results in bold.

perfect accuracy for the broader semantic group classification (4 classes). ComplEx, DistMult, and SimplE perform slighty worse, Snomed2Vec slightly below them, and Cui2Vec falls behind by a significant margin. We see a greater discrepancy in relative performance by model type in semantic type classification (32 classes), in which more fine-grained semantic information is required.

Two advantages of the semantic type and group entity classification tasks are: (i) information is provided by the UMLS, making the task non-proprietary and standardized; (ii) it readily shows whether a model preserves the semantic structure of the ontology, an important aspect of the data. The tasks can also easily be modified for custom data and specific domains, e.g. class labels for genes and proteins relevant to a particular biomedical application can be used in classification to assess how well the model captures relevant domain-specific information.

For comparison to related work, we also examine the benchmark tasks to assess medical concept embeddings based on *statistical power* and *cosine similarity bootstrapping*, proposed by [17]. For a given known relationship pair (e.g. x `cause_of` y), a null distribution of pairwise cosine similarity scores is computed by bootstrapping 10,000 samples of the same semantic category as x and y respectively. The cosine similarity of the observed sample is compared to the 95th percentile of the bootstrap distribution (statistical significance at the 0.05 level). The authors claim that, when applied to a collection of known relationships (causative, comorbidity, etc), the procedure estimates the fraction of true relationships discovered given a tolerance for some false positive rate. Following this, we report the statistical power of all 7 models for two of the tasks: *semantic type* and *causative relationships*. The former (**ST**) aims to assess a model's ability to determine if two concepts share the same semantic type. The latter consists of two relation types: `cause_of` (**Co**) and `causative_agent_of` (**CA**). Results are reported in Table 4.3. The cosine similarity bootstrap results, particularly for the causative relationship tasks, illustrate a major flaw in the protocol. While Snomed2Vec and Cui2Vec attain similar statistical powers for **CA** and **Co**, we see large discrepancies between the two tasks for the KGE models, especially for ComplEx, DistMult, and SimplE, which produce globular embedding clusters. Examining the dataset, we observe that the `cause_of` relations occur mostly between concepts *within* the same semantic group/cluster (e.g. Disorder), whereas the `causative_agent_of` relations occur between concepts in *different* semantic groups/clusters (e.g. Chemicals to Disorders). The large discrepancy in **CA** task results for the KGE models is because using cosine similarity embeds the assumption that all related entities are close, regardless of the relation type. The assumption that cosine similarity in the concept embedding space is an appropriate measure of a diverse range of relatedness (a much broader abstraction that subsumes semantic similarity and causality), renders this evaluation protocol unsuitable for assessing a model's ability to capture specific types of relational information in the embeddings. Essentially, all that can be said about the cosine similarity-based procedure is that it assesses how close entities are in that space as measured by cosine distance. It does not reveal the nature of their relationship or what kind of relational information is encoded in the space to begin with.

In contrast, KGE methods explicitly model relations and are better equipped to make inferences about the relational structure of the knowledge graph embeddings.

Thus, we propose *relation prediction* as a standard evaluation task for assessing a model's ability to capture information about relations in the knowledge graph. We simply modify the link prediction task described above to accommodate `relation` as a target (formulated as (h, ?, t), generating ranking-based metrics for the model's ability to prioritize the correct relation type given a pair of concepts. This provides a more principled and interpretable way to evaluate the models' relation representations directly based on the model prediction. The last 3 columns of Table 4.3 report relation prediction metrics for the 5 KGE models. In particular, RotatE and SimplE perform well, attaining around 0.8 Hits@1 and around 0.85 MRR.

We conduct error analysis to gain further insight by categorizing relation types into 6 groups based on the *cardinality* and *homogeneity* of their source and target semantic groups. If the set of unique head or tail entities for a relation type in the dataset belongs to only one semantic group, then it has a cardinality of 1, and a cardinality of `many` otherwise. If the mapping of the source semantic groups to the target semantic groups are one-to-one (e.g. DISO to DISO and CHEM to CHEM), then it is considered homogeneous. We report relation prediction metrics for each of the 6 groups of relation types for RotatE and ComplEx in Table 4.4.

We see that RotatE gives impressive relation prediction performance for all groups except for many-to-many-homogeneous, a seemingly challenging group of relations containing ambiguous and synonymous relation types, e.g. `possibly_equivalent_to`, `same_as`, `refers_to`, `isa`. In contrast, ComplEx struggles with a wider array of relation types, suggesting that it is generally less able to model different types than RotatE. The last two rows under each model show per-relation results for the causative relationships mentioned previously: `cause_of` and `causative_agent_of`. RotatE again shows significantly better results compared to ComplEx, in line with its theoretically superior representation capacity [150].

4.7 Discussion

Based on our findings, we recommend the use of KGE models to leverage the multi-relational nature of knowledge graphs for learning biomedical concept and relation embeddings; and of appropriate evaluation tasks such as link prediction, entity classification and relation prediction for fair comparison across models. We also encourage analysis beyond standard validation metrics, e.g. visualization, examining model predictions, reporting metrics for different relation groupings and devising problem or domain-specific validation tasks. A further promising evaluation task is the triple prediction proposed in [6], which we leave for future work. A more ideal way to assess concept embeddings in biomedical NLP applications and patient-level modeling would be to design a suite of benchmark downstream tasks that incorporate the embeddings, but that warrants a rigorous paper of its own and is left for future work.

We believe this paper serves the biomedical NLP community as an introduction to KGEs and their evaluation and analyses, and also the KGE community by providing a potential standard benchmark dataset with real-world relevance.

4.8 Conclusion and Future Work

We present results from applying 5 leading KGE models to the SNOMED-CT knowledge graph and compare them to related work through visualizations and evaluation tasks, making a case for the importance of using models that leverage the multi-relation nature of knowledge graphs for learning biomedical knowledge representation. We discuss best practices for working with biomedical knowledge graphs and evaluating the embeddings learned from them, proposing link prediction, entity classification, and relation prediction as standard evaluation tasks. We encourage researchers to engage in further validation through visualizations, error analyses based on model

Relation	MRR	H@1	H@10	Count
ComplEx				
1-1-hom	.600	.319	.944	72
M-M-hom	.605	.417	.877	29,028
M-1	.683	.557	.884	2,509
1-M	.738	.640	.916	2,497
1-1	.889	.817	.995	420
M-M	.867	.819	.941	15,044
Co	.706	.662	.779	145
CA	.857	.822	.908	303
RotatE				
M-M-hom	.784	.718	.934	29,028
M-M	.973	.944	.992	15,044
M-1	.971	.945	.998	2,509
1-M	.975	.953	.998	2,497
1-1	.985	.959	1.	420
1-1-hom	.972	.976	1.	72
Co	.803	.738	.890	145
CA	.996	.993	1.	303

Table 4.4: Relation prediction results for RotatE and ComplEx by category of relation type (last two rows relate to causative relation types).

predictions, examining stratified metrics, and devising domain-specific tasks that can assess the usefulness of the embeddings for a given application domain.

There are several immediate avenues of future work. While we focus on the SNOMED-CT dataset and the KGE models implemented in GraphVite, other biomedical terminologies such as the Gene Ontology [153] and RxNorm [118] could be explored and more recent KGE models, e.g. TuckER [14] and MuRP [15], applied. Additional sources of information could also potentially be incorporated, such as textual descriptions of entities and relations. In preliminary experiments, we initialized entity and relation embeddings with the embeddings of their textual descriptors extracted using Clinical Bert [8], but it did not yield gains. This may suggest that the concept and language spaces are substantially different and strategies to jointly train with linguistic and knowledge graph information require further study. Other sources of information include entity types (e.g. UMLS semantic type) and paths, or multi-hop generalizations of the 1-hop relations (triples) typically used in KGE models [71]. Notably, CoKE trains contextual knowledge graph embeddings using path-level information under an adapted version of the BERT training paradigm [163].

Lastly, the usefulness of biomedical knowledge graph embeddings should be investigated in downstream applications in biomedical NLP such as information extraction, concept normalization and entity linking, computational fact checking, question answering, summarization, and patient trajectory modeling. In particular, entity linkers act as a bottleneck between text and concept spaces, and leveraging KGEs could help develop sophisticated tools to parse existing biomedical and clinical text datasets for concept-level annotations and additional insights. Well performing entity linkers may then enable training knowledge-grounded large-scale language models like KnowBert [126]. Overall, methods for learning and incorporating domain-specific knowledge representation are still at an early stage and further discussions are needed.

Chapter 5

Incorporating Domain Knowledge Into Language Models Using Graph Convolutional Networks for Clinical Semantic Textual Similarity

5.1 Introduction

Electronic health records (EHR) have introduced efficiencies in clinical documentation with the automatic insertion of commonly used documentation phrases and the "copy-and-paste" command that copies the content of one day's note into that of the next, but at the same time, these tools have led to notes becoming increasingly bloated with sometimes outdated, irrelevant, and even erroneous information [79].To trim down bloated clinical documentation, one approach of interest is to identify highly similar text snippets for the goal of removing such text; Wang et al created the MedSTS dataset, a clinical analogue of the natural language understanding benchmark task called semantic textual similarity (STS), to be a resource for this line of study. In this workshop paper, we show the model, as well as subsequent improvements, used in the August 2019 National NLP Clinical Challenges (n2c2) / Open Health NLP Consortium (OHNLP) semantic similarity shared task challenge which featured the MedSTS dataset. In the broader natural language processing (NLP) community, STS assessment is a task to calculate the similarity of semantic meaning and content between natural language texts [164], and at the time of its release in late 2018, the BERT (Bidirectional Encoder Representations from Transformers) language model had the best published performance on the commonly used general English Semantic Textual Similarity Benchmark, known as STS-B [55]. For the MedSTS dataset, it was shown that a BERT model fine–tuned to the biomedical domain also outperformed most prior state-of-the-art models [121]. The first iteration of the MedSTS challenge in 2018, prior to the release of BERT, saw four submissions with a mixed use of traditional machine learning models like random forests and more recent deep learning architectures like recurrent neural networks (RNNs) and convolutional neural networks (CNNs). The 2019 MedSTS challenge saw over thirty submissions, with the majority of them using BERT in some capacity. The increased number of submissions as well as the increased average performance of those submissions can be attributed in large part to the recent progress in language models, of which BERT is a popular example.

Despite such advances, researchers have noted that although language models demonstrate a small degree of common sense reasoning and basic knowledge, such models have very limited ability to generate factually correct text or even recall explicit facts in the training data [5]. Attempts to mitigate such shortcomings of language models have often involved the use of graph representation learning techniques [126, 185, 109], which provide a natural way to work with knowledge in the form of graphs.

Recent progress in graph representation learning has given rise to two promising classes of methods that could be used in conjunction with NLP models to incorporate knowledge (either domain knowledge or commonsense knowledge): graph convolutional networks (GCN) [94] and knowledge graph embeddings (KGE) [65].

GCNs generalize the notion of convolution from images to graph–structured data, enabling the application of deep learning techniques on graphs. KGE methods encode entities (nodes) and relationships (edges) in a knowledge graph into dense vector representations, much like word embeddings. KGEs provide a way to obtain embeddings of concepts, and GCNs are a natural way to use that information in the context of graph–based learning, for instance by initializing the node features with pretrained KGEs. In this paper, we leverage these recent advances in NLP and graph representation learning to develop a more knowledge–aware approach to the MedSTS benchmark dataset. We further investigate the benefits of other techniques such as data augmentation, multi–source ensembling, and knowledge distillation and obtain competitive performance for the task as of 2019.

5.2 Methods

5.2.1 Dataset

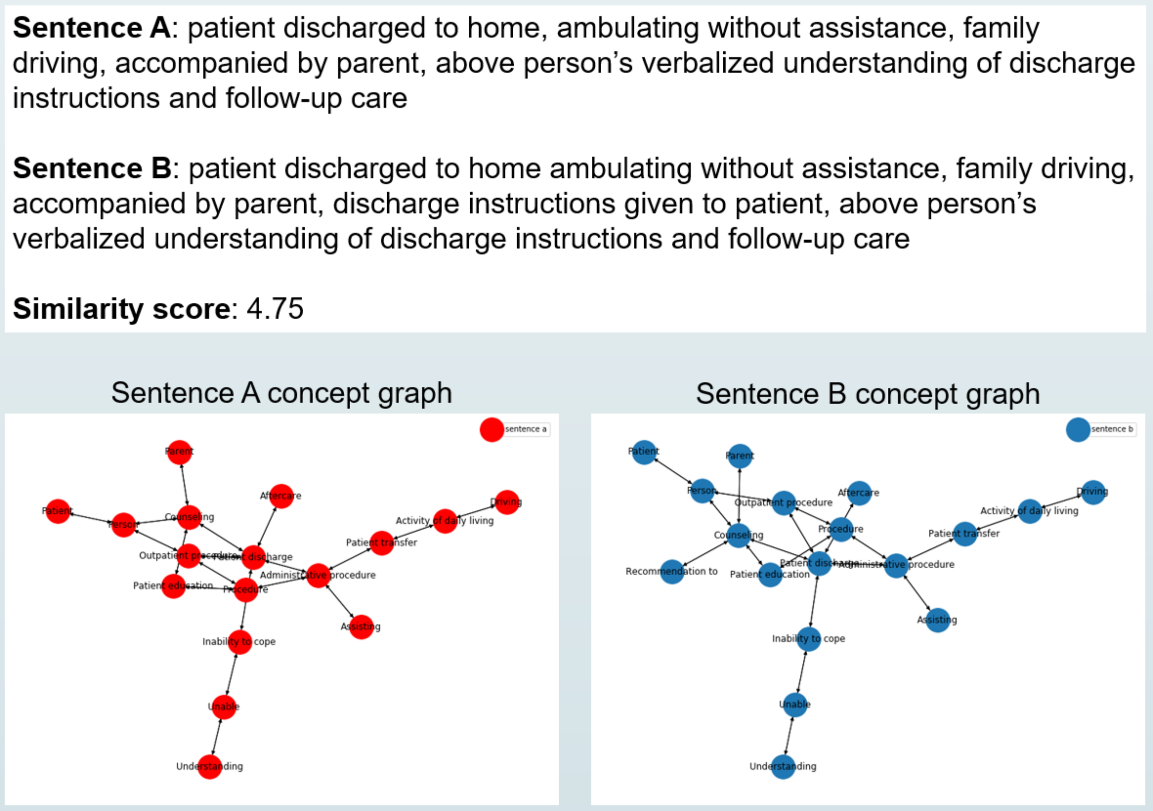
MedSTS is a dataset of sentence pairs gathered from the clinical electronic health records at Mayo Clinic. Deidentified sentences were selected on frequency of appearance based on an assumption that frequently appearing sentences tend to contain less protected health information. Sentence pairings were arranged to have at least some degree of surface level similarity based on a combination of surface lexical similarity metrics. Broadly speaking, sentences generally fell into four categories: signs and symptoms, disorders, procedures, and medications. Further details are discussed in the original MedSTS paper [164]. For the 2019 n2c2 / OHNLP competition and this study, a subset of annotated sentence pairs was examined; there were 1642 sentence pairs (80%) in the training set and 412 pairs (20%) in the test set [165]. This subset was independently scored by two medical experts for semantic similarity. A 6 point (0-5) rubric was provided to the annotators where 0 denotes complete dissimilarity, 1 indicates that two sentences are topically related but otherwise not equivalent, and 5 represents complete similarity. The agreement between the two annotators received a weighted Cohen's Kappa score 0.67. The average of the two scores served as the gold standard against which STS systems would be evaluated.

5.2.2 Concept graph construction

For each sentence in the MedSTS dataset, we constructed a corresponding concept graph of the sentence to represent the domain knowledge aspect of the dataset. The concept graph consists of concepts that were tagged with a domain-specific tagger called MetaMap [10] and mapped to a specified medical terminology. The idea is that such a graph provides an additional representation of the data that contains explicit domain knowledge in the form of mapped concepts and their connections.

The Unified Medical Language System (UMLS) [22] is an important resource in biomedical and healthcare research that integrates many health and biomedical vocabularies and terminologies under a unified, interoperable system. MetaMap is a widely used NLP tool that maps concepts in biomedical and clinical text to the UMLS Metathesaurus. We apply MetaMap on the MedSTS dataset to extract biomedical and clinical entities belonging to the SNOMED-CT terminology under the UMLS. Thus, for each sentence, we obtain a corresponding list of extracted concepts, their concept unique identifiers (CUIs), and semantic type information.

Then we construct a graph of SNOMED-CT terminology from the raw UMLS files with the concepts (MRCONSO.RRF) as nodes and the relationships (MRREL.RRF) between them as edges. For simplicity, we only consider the connectivity information between the concepts and leave the semantic information in the relation types to future work. Once we have a full SNOMED graph, we induce subgraphs for each sentence from MedSTS by taking the shortest paths between the concepts extracted from the sentence. More concretely, this is done using the shortest path method with the Dijkstra algorithm in the Networkx [73] library. While there are many possible ways of constructing such sentence graphs, we stick to the simple heuristic of shortest paths to obtain a connected graph representing each sentence. Examples of such concept graphs along with their original sentences are shown in Figure 5.1.



Sentence A: patient discharged to home, ambulating without assistance, family driving, accompanied by parent, above person’s verbalized understanding of discharge instructions and follow-up care

Sentence B: patient discharged to home ambulating without assistance, family driving, accompanied by parent, discharge instructions given to patient, above person’s verbalized understanding of discharge instructions and follow-up care

Similarity score: 4.75

Figure 5.1: An illustration of sentence graphs constructed from a pair of sentences with a similarity score of 4.75.

5.2.3 Data augmentation

Given the small size of the dataset, we decided to augment it by including additional domain knowledge from the MetaMap output files. Notably, there are two pieces of information we chose to use: the preferred name of the mapped concept in the source terminology and the semantic type of the concept within the UMLS Semantic Network. The preferred name of a mapped concept can often be the same as it appears in the text, but sometimes it provides potentially valuable information in the form of synonyms or abbreviation expansion. For example, in the text snippet "the patient was taken to the pacu in stable condition", the term "pacu" is mapped to the UMLS concept "Postoperative anesthesia care unit (PACU)", providing the full description of the abbreviated term. The strings of the preferred names of mapped concepts are simply appended to the original sentences in the dataset. Likewise, the semantic types of the mapped concepts ("Health Care Related Organization" for the term "pacu") are appended to the original sentences. Another trick we used was to double the dataset size by simply feeding the model a copy of the dataset with the sentences in reverse order (i.e. "sentence2:sentence1"), which yielded slightly better results than simply doubling the number of training epochs, suggesting that showing the model the reverse copy of the dataset might give it more explicit hints that the task is agnostic to the ordering of the sentences. While the data augmentation techniques we used are simple and yield moderate improvements in performance, we refer to a recent paper [167] for more interesting approaches to data augmentation in which they use back-translation and segment reordering to augment the MedSTS dataset.

5.2.4 BERT

BERT is a widely used NLP model that is among the recently emerging class of language models that use transformers [158] as the building blocks, stacking multiple layers of transformer-based modules that primarily use the multi-headed self-attention mechanism to encode text into dense embeddings. The model is trained using the masked language modeling objective and the next sentence prediction objective, and pretrained models for BERT (and other similar models) are readily available on the

Huggingface Transformers library [171]. Shortly after BERT dominated the general NLP field, several variations of BERT adapted to the biomedical and clinical domains also became available [121? , 101]. These domain–adapted versions of BERT were trained on some combination of MIMIC-III [90], PubMed, and Pubmed Central, and have been shown to outperform the original BERT model on several clinical NLP tasks, suggesting that they are more appropriate for working with clinical text datasets like MedSTS.

5.2.5 Graph convolutional networks

Kipf et al. contributed to the popularization of graph neural networks by providing an efficient implementation of GCN and demonstrating its effectiveness on several benchmark graph datasets for graph classification, node classification, and link prediction [94]. Variants of GCNs were soon applied successfully to various domains and problems, including modeling interactions in physical systems [92], drug-drug interactions [192], and text classification [180]. GCNs have become a popular deep learning model for working with graph–structured data, and we use GCNs to encode the concept graphs.

5.2.6 Knowledge graph embeddings

KGEs are a relatively novel class of methods for learning dense vector representations of entities and relations in multi-relational, heterogeneous knowledge graphs. Essentially, a KGE model maps entities and relations to embedding spaces using a predefined scoring function. Due to their growing popularity and availability of implementation, KGEs have recently been applied to various domains including biomedical knowledge graphs [33]. Chang et al. show that using KGEs for learning concept embeddings from medical terminologies and knowledge graphs is arguably a more principled and effective approach than previous methods based on skip-gram based models like Cui2Vec [17] or network embedding–based models like Snomed2Vec [3]. While we initially used Cui2Vec for our entity vectors at the time of submission, we later used Snomed KGE after it became available in recent months.

5.2.7 Augmenting BERT with KGEs for MedSTS

We combine the above components into a single model in the following way: we use a BERT-based model as our text encoder for the sentence pairs in MedSTS, use a GCN–based model as our graph encoder for the concept graphs corresponding to the sentence pairs, initialize the node embeddings in the graphs with pretrained Snomed KGEs, concatenate the outputs of the text and graph encoders, and pass the final concatenated vector to a fully connected layer to obtain the semantic similarity score. We also test the benefits of using the Snomed KGEs by comparing it with random initialization and initializing with Cui2Vec embeddings. A visualization of the pipeline is shown in Figure 5.2.

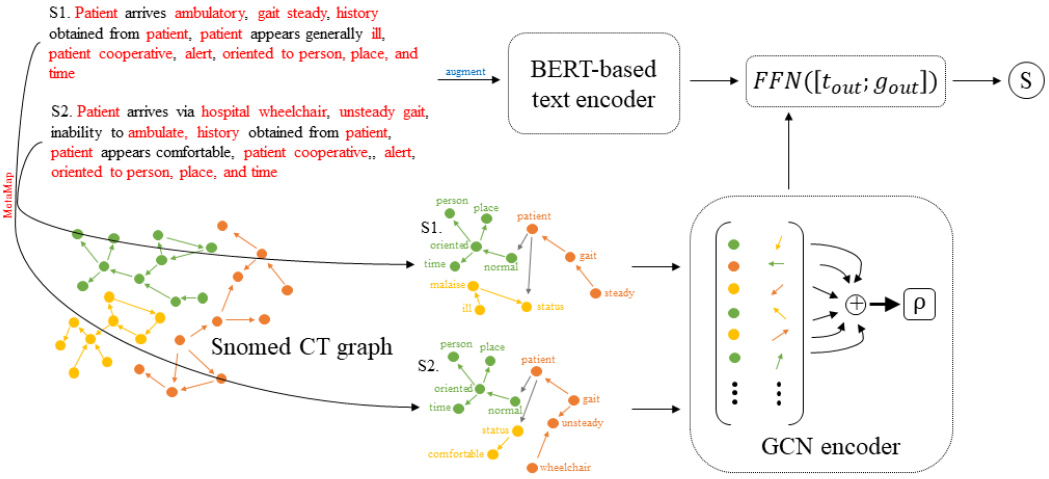

S1. Patient arrives ambulatory, gait steady, history obtained from patient, patient appears generally ill, patient cooperative, alert, oriented to person, place, and time
S2. Patient arrives via hospital wheelchair, unsteady gait, inability to ambulate, history obtained from patient, patient appears comfortable, patient cooperative,, alert, oriented to person, place, and time
MetaMap
augment
BERT-based text encoder
$FFN([t_{out}; g_{out}])$
S
Snomed CT graph
S1.
person
place
patient
oriented
time
normal
gait
malaise
steady
ill
status
S2.
person
place
patient
oriented
time
normal
gait
status
unsteady
comfortable
wheelchair
⊕
ρ
GCN encoder

Figure 5.2: A visualization of the pipeline that shows the graph construction process and the model combination.

5.2.8 Ensemble and knowledge distillation

After training our model, we take an ensemble to further improve the performance. Following [176], we do multi-source ensembling with the following variants of BERT: BERT-base [55], SciBERT [19], ClinicalBERT [8], MT-DNN (Multi-Task Deep Neural Networks) [110], and BlueBERT [121]. Then we perform knowledge distillation, which is an effective model compression method in which a smaller model is trained to mimic a larger model (i.e. the ensemble). We use the predictions of the multi-source ensemble model as soft labels in a teacher bounded regression loss function following [37] to train more individual models, then obtain a final ensemble of the knowledge–distilled models.

5.3 Results

We split the provided training set of MedSTS into 1313 training examples and 329 validation examples and report the Pearson correlation for the held-out test set of 412 examples; Pearson correlation was the chosen metric for the competition. We used the Huggingface Transformers library for implementations related to language models, and we used Pytorch Geometric [59] for implementations of GCNs. Much of the default training and fine–tuning hyperparameters were used while the following hyperparameters were tuned on the validation set: learning rate of 1e-4 for BERT-based models (from [5e-5, 1e-4, 5e-4]), learning rate of 1e-3 for GCNs (from [1e-2, 1e-3, 1e-4]), and the number of epochs of 4 (from [3, 4, 5]).

Table 5.1 shows the contributions of the different components in the pipeline. Simply using BERT-base off the shelf and fine–tuning it on MedSTS yields higher performance compared to the 2018 submissions. Using ClinicalBERT and our data augmentation technique described above each yield moderate gains.

Adding a graph encoder on top of that to incorporate the concept graphs showed minor improvements when the node embeddings were initialized either randomly or with pretrained Cui2Vec embeddings. However, using Snomed KGE as the node features in the GCN resulted in an increase of 1.3 points over just ClinicalBERT with data augmentation, suggesting that Snomed KGE serves as a better starting representation of the concepts. It's worth noting that since the BERT–based text encoder is initialized with a pretrained checkpoint, it might be especially important to initialize the graph encoder with decent pretrained embeddings to allow it to "catch up" to the text encoder. We call this best performing setting ClinicalBERT-all.

We also manually categorized the sentence pairs into four categories: sentences relating to patient condition and status (status), education or interaction with patient (education), medication (meds), and miscellaneous or clearly dissimilar topics (misc). The columns in Table 5.1 under Pearson correlation show the scores for the test set (all) and for the four categories described. Sentence pairs in the status and education categories received relatively higher scores, as expected since many of the sentences and text snippets in these categories are often repeated. Specifically, text snippets beginning with "patient arrives . . . ", "discussed the risks . . . ", and "identified illness as a learning need . . . " recur noticeably in the two categories. Further, the medication and miscellaneous categories received relatively low correlation scores. For the miscellaneous category, this is expected since many of the sentence pairs in this category are more difficult for the model to learn due to their greater variability. For the medication category, the gold standard scores assigned by the annotators proved to be rather inconsistent and challenging to predict even upon manual review by a medical expert.

Table 5.2 shows the results for ensembling and knowledge distillation. First, we took the ensemble of 10 ClinicalBERT-all with slightly varied hyperparameters and saw a moderate increase in performance, as expected of ensembles. Then, following [110], we took an ensemble of 10 models consisting of a variety of model types (BERT-base,

Table 5.1: Results for the base model and each version with an additional component added to the system. DA refers to data augmentation. The columns under Pearson correlation show the scores for the test set (all) and four subsets of the test set that include sentences regarding patient condition or status (status), education or interaction (education), medication (meds), and miscellaneous topics (misc.).

Model	Pearson correlation.				
	all	status	education	meds	misc.
BERT_base	0.842	0.643	0.721	0.522	0.414
ClinicalBERT	0.848	0.662	0.735	0.541	0.425
ClinicalBERT-DA	0.855	0.671	0.737	0.553	0.432
ClinicalBERT-DA + GCN_rand	0.861	0.675	0.742	0.532	0.427
ClinicalBERT-DA + GCN_cui2vec	0.863	0.682	0.753	0.536	0.442
ClinicalBERT-DA + GCN_snomedkge	0.868	0.693	0.761	0.562	0.463

Table 5.2: Results for ensembling the best performing model from (Table 1) (ClinicalBERT-all), ensembling with multiple language models (LM) each with a graph convolutional network (GCN), and ensembling of knowledge distilled (KD) multi-source ensembles. The best performing model from the IBM team at the time of the competition is included for reference.

Ensemble Type	Pearson corr.
Ensemble of ClinicalBERT_all	87.5
Ensemble with multiple LMs	87.8
Ensemble of KD models	88.2
IBM-N2C2	90.1

SciBERT, ClinicalBERT, MT-DNN, BlueBERT) along with the graph encoder based on their validation performance and saw a slight improvement. Finally, using teacher bounded regression loss [37], we used the outputs of the multi-source ensemble model as soft labels to train more best-setting models of different types, and took an ensemble consisting of 10 such knowledge–distilled models for slight performance gain.

5.4 Discussion

5.4.1 Main findings

We implemented a list of techniques in our pipeline for the MedSTS clinical semantic textual similarity benchmark task and reported slight to moderate improvements in performance for each. Using a pretrained BERT-based model off-the-shelf and fine-tuning it alone serves as a strong baseline that outperforms all pre-BERT systems for the task. We find that our data augmentation technique helps slightly, but again we refer to [167] to more interesting and effective data augmentation approaches for MedSTS.

Adding a graph encoder to incorporate concept graphs into the pipeline yielded decent gains, especially when the graph encoder was initialized using pretrained Snomed KGE. We stress that since the graph encoder is trained jointly with a pretrained text encoder, it is important to consider providing it with pretrained embeddings as well so that it doesn't fall too far behind in training. As expected, ensembling leads to improved performance, and further improvements can be achieved by leveraging multiple sources of language models as well as knowledge distillation followed by another ensembling of the distilled models.

We also attempted several other techniques that did not yield any performance gains. First, we tried multi-task learning using different general and clinical domain NLP datasets including MedNLI [134], RQE [21], and STS-B [31] following an implementation of multi-task learning for MT-DNN, but this approach did not lead to any improvements while significantly increasing training time. Second, we tried manually annotating the MedSTS data for different sentence categories (Medication, Status, Education, and Miscellaneous) to use as an auxiliary classification task (also an example of multi-task learning), but it did not lead to noticeable gain in performance. Lastly, we tried experimenting with different variants of GCNs, but we found that training multiple types of graph neural networks jointly with a large language model is difficult in terms of hyperparameter tuning and decided to limit our analysis to basic GCNs.

5.4.2 Limitation of method

While the results demonstrate that the strategies for data augmentation and incorporating domain knowledge through concept embeddings and GCNs do confer some benefit, we address some of the limitations in this section.

The data augmentation techniques we used involve including additional textual and semantic information from the MetaMap output and reversing the sentence ordering to double the dataset size. There are many other potential data augmentation techniques in the general NLP field that could be useful. Notably. Wang et al. [167] recently used segment reordering and back translation to significantly improve their model performance on the task.

As for the pretrained concept embeddings and GCNs, combining them with a large pretrained language model is still largely experimental and could be improved by utilizing recent developments in the field of graph representation learning such as Graph Attention Networks [159] and Graph Matching Networks [104].

5.4.3 Limitation of dataset

Both the positive and negative findings should be considered with caution due to the abundance of potential ways of implementing each component as well as the relatively small size and limited quality of the dataset as compared to mainstream non-clinical NLP domains that have less complicated access to labeled data.

After working closely with the dataset for several months, we noticed that certain sentence pairs had large irregularities in scoring from the two annotators of the dataset. This was most notable in the sentence pairs that discussed medications; often these sentence pairs describe the prescribing of medications to patients and differed on dosing or drug class. At one level of categorization, the similarity of a sentence pair related to prescribing could be seen as high regardless of the medication class or dosing. At another level of categorization, it appeared that several of such pairs were noted to be of low similarity when the medications or dosing regimens differed; this discrepancy in scoring also seemed to differ depending on the drug classes being mentioned. Without knowing which annotator was behind a given score, it is difficult to speak conclusively, but we speculate that certain drug classes were of greater salience to each annotator. As an example, someone in a mental health specialty may subjectively perceive two different psychiatric medications of different classes to be quite different but view cardiology drugs to be subjectively more similar. In contrast, an individual in the field of cardiology may perceive various cardiology drugs as being different whereas psychiatric medications as a category may seem overall more similar. Such differences in perspective may also be influenced by aspects of the annotator's practice—whether their practice occurs in inpatient settings or outpatient settings, the operating room or the medical clinic.

Much of the scoring irregularities may be related to the nature of the task in rating subjective similarity. One approach to mitigate annotator bias as discussed in the original MedSTS paper is to increase the number of annotators and set the average score as the gold standard. For example, in STS-B, 5 annotators were used for each sentence, and annotators were limited to the number of sentence pairs that they could annotate. While such an approach could be prohibitively expensive to hire enough medical annotators and very cumbersome to implement for clinical text given patient privacy protections, another approach in the case of having few annotators could be to reveal potentially biasing factors, such as clinical background, towards annotation or assign an annotator ID behind each scoring. Stating the biases or allowing teams to model the annotator biases may help with understanding scoring irregularities which may be difficult to resolve without significantly tailored algorithm designs or features that require specific domain knowledge to adapt to unique annotator biases. Despite our concerns with the fundamental difficulty with objectively rating subjective semantic similarity, the high Pearson correlation demonstrated by our model suggests that the task is still largely tractable. MedSTS also remains one of the few, if not only, publicly available datasets for studying clinical semantic textual similarity for EHRs. We hope that our suggestions may introduce additional strategies to model the variance from subjective elements and provide some insights for future dataset annotation processes for this important yet challenging problem.

5.5 Conclusion

As participants of the 2019 n2c2 / OHNLP shared task challenge, we developed a system for the MedSTS clinical semantic textual similarity benchmark task by combining BERT–based text encoders and GCN–based graph encoders in order to incorporate domain knowledge into the NLP pipeline. We also experimented with other techniques involving data augmentation, pretrained concept embeddings, ensembling, and knowledge distillation to further increase our performance. Though the results lagged behind the top scoring model at the n2c2 workshop, the incorporation of domain knowledge via graph-based methods into deep learning NLP models was a new advance in clinical NLP. We highlight our concerns about the impact of specific difficulties with subjective semantic similarities in dataset annotation, but overall, we believe that clinical semantic similarity remains an important topic of study and continued work on the MedSTS benchmark, one of the few clinical semantic textual similarity datasets available, will yield advances in processing valuable unstructured data in EHRs. The MedSTS dataset should continue to be improved and enlarged through further careful annotation of the original pool of sentence pairs, and future work should explore novel methods that can effectively leverage both linguistic and domain knowledge.

Chapter 6

OREO: Multimodal Patient Representation with Transformers

6.1 Abstract

Transformer-based language models have become prevalent in the biomedical and clinical natural language processing literature in recent years. However, integration of text with other modalities of patient data (i.e. diagnoses, medications, and lab tests) has been lacking. Existing methods for multimodal patient representation learning mostly focus on different types of medical codes, and ones that attempt integrating text with codes typically involve simple concatenation of separate modality representations. Considering the inherent heterogeneity of clinical data as well as the importance of domain knowledge, integrative models that can effectively perform multimodal learning with text and structured knowledge must be explored. In this paper, we propose a simple and extensible approach to multimodal learning for clinical data within the widely used framework of transformer-based language models. This approach consists of vocabulary expansion and a graph representation module that leverages latent structural information in the multimodal input. We demonstrate its

effectiveness through different tasks and settings and provide in-depth discussions of important issues in the field that should be addressed. The code will be available on github.

6.2 Introduction

Transformer-based language models (LMs) [55] have become a standard part of the biomedical and clinical natural language processing (NLP) toolkit, raising the baseline performance across a wide range of tasks [102] and leading to subsequent domain-specific models pretrained on domain-specific corpora [8, 169].

These domain-specific LMs have successfully leveraged transfer learning to better capture linguistic information relevant to particular domains based on the training corpora. While these models, trained mostly using the masked language modeling (MLM) pretraining objective on domain-specific corpora, are able to capture some domain knowledge in a distributed sense based largely on co-occurrence information, this approach is data inefficient and provides little control over what is actually learned.

Considering the importance of domain knowledge in clinical and biomedical settings, as well as the fact that a lot of knowledge is encoded in symbols (i.e. codes) as part of some structured terminology or ontology (e.g. ICD for diagnosis, CPT for procedures, RxNorm for medications, UMLS for general medical concepts, etc), the incorporation of structured knowledge into the current family of transformer-based LMs is a topic that should be further explored.

Models that can learn the representation of language and structured knowledge jointly are a crucial step toward expanding our capacity to fully leverage the information stored in clinical and biomedical databases. Such models would fall under multimodal learning, a challenging open area of research that has been getting increasing attention, both in biomedical NLP and broader machine learning.

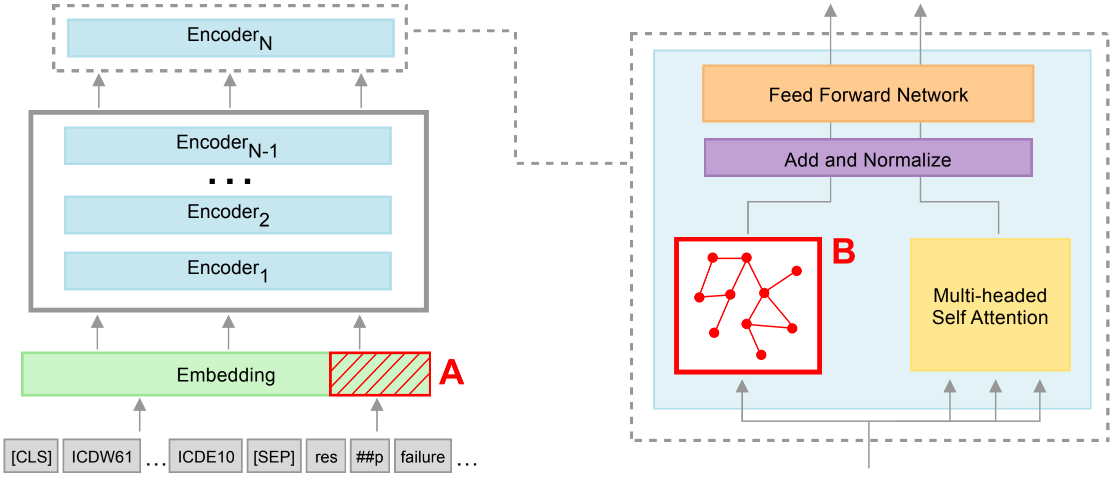


Figure 6.1: OREO extends a transformer-based LM architecture by (A) extending the vocabulary to include ICD codes in the embedding layer and (b) including a graph learning module that is trained simultaneously with self-attention in the final Encoder layer.

So far, existing methods in patient representation learning have mostly focused on either only text or only structured information such as medical codes, and attempts at multimodal learning involving both text and medical codes typically concatenate the representations of separate modalities to obtain a combined representation.

Further, existing methods for multimodal patient representation learning are cumbersome to incorporate into the widely used transformer-based LM framework and thus limited in their usefulness.

In this paper, we propose a method that incorporates structured domain knowledge seamlessly within the existing framework of transformer-based LMs as well as a graph learning module that leverages the latent structural information within the multimodal patient data to enhance the learned representation. This method is straightforward to implement in any of the existing transformer-based LMs and is extensible to multiple modalities that can be represented as symbols or tokens. The graphical abstract of the approach is shown in Figure 6.1.

6.3 Background

Since the initial release of pretrained BERT models [55], several domain-specific variants of transformer-based LMs have been trained on biomedical and clinical text corpora [101, 8], typically some combination of PubMed abstracts[1], PubMed Central full articles[2], and MIMIC-III clinical notes [90].

The capacity of these LMs to effectively learn from biomedical text has solidified their place in the biomedical NLP literature, with works involving transformer-based LMs accounting for a significant portion of the recent BioNLP and Clinical NLP workshop proceedings [52, 139].

While recent progress in NLP has greatly enhanced our ability to handle clinical text, it must be contextualized within the broader objective of learning useful patient representations from EHR data, which is inherently heterogeneous and also includes diagnosis codes, lab orders and results, medications, and so on.

Under the premise that patients can be represented as sequences of medical codes, NLP methods have frequently been applied to non-textual EHR representation learning with promising results [170].

Models such as Deepr [144] and RETAIN [42] use RNN-based architectures to produce clinical concept embeddings that take into account visit- and patient-level information. [147] train an RNN-based language model on patients' diagnoses, procedures, medications, and laboratory tests in the form of their respective codes (ICD10, CPT, RXCUI, and LOINC), but they do not use any text from clinical notes.

[98] also represent patients as sequences of medical codes, and instead of using clinical text, they use an annotation tool to extract clinical concepts from the text, indirectly making use of the notes in the form of concept codes that are used to train their ConvAE model to derive patient embeddings.

[1] https://pubmed.ncbi.nlm.nih.gov/

[2] https://www.ncbi.nlm.nih.gov/pmc/

Other methods, motivated by the success of the transformer in NLP, have adapted the model for clinical tasks. [182] use self-attention modules to capture the multilevel structure of medical codes, visits, and patients along with their temporal information. BEHRT [103] modifies the BERT architecture to train on sequences of diagnosis codes, instead of text, for all visits pertaining to a patient to predict diagnoses in future visits.

More recent works have attempted to improve patient embeddings by leveraging the latent structure of EHRs, including hierarchical relationships between treatments and conditions [43] and visits and diagnoses [78]. This latent structure can also be learned as part of the training process in methods like the Graph Convolutional Transformer (GCT) [44] and the end-to-end latent graph learning approach proposed by [49].

Whether this latent data structure is learned, manually constructed, or given, it can be represented as a graph and passed to a graph convolutional network (GCN) [94] to integrate multiple modalities. Some recent examples of GCN-based multimodal learning with clinical data include [133] (physiological time-series and diagnoses) and [100] (diagnoses and medical concepts).

There are two main limitations with existing methods described so far. First, they have not integrated free-text directly with medical codes, resorting to simple concatenation toward the end of training, which hinders their ability to leverage relationships between the different modalities. Second, these methods are cumbersome to integrate into current transformer-based LM frameworks. While embeddings could be separately obtained using available implementations of these existing methods and incorporated into the transformer-based LM pipeline through concatenation, an end-to-end design within the currently dominant framework would drastically improve the usability of such methods.

In this paper, we propose a simple and extensible approach to incorporate structured knowledge into the transformer-based LM framework as well as an additional graph representation module that further enhances the jointly learned representation of the multimodal inputs by leveraging their latent structure.

We also offer discussions regarding important topics like tokenization, limited context windows, and latent graph learning with directions for future work.

6.4 Methods

6.4.1 Datasets and Tasks

We use the MIMIC-III dataset [90] for two tasks: in-patient mortality and 30-day readmission.

The in-patient mortality prediction dataset was prepared following [169]. Notes written by physicians and nurses at least 24-hours before discharge were included.

Around 10% of patients expire at the end of an admission, and we balance positive and negative examples by sampling. Notes longer than 365 words based on simple whitespace splitting are divided into multiple chunks, preserving corresponding visit-level labels and ICD-9 codes. The final chunked dataset has 423,618 samples (35,482 patients) and is split into train, validation, and test sets in a roughly 75:10:15 ratio. The average number of samples per visit is 6.48 before chunking and 9.98 after chunking. The average number of codes per visit is 13.7.

The 30-day readmission prediction dataset was prepared following [86]. Only the last discharge summary for each visit was included, and the final chunked dataset has 27,153 samples (6,163 patients) split roughly into 80:10:10. The average number of samples per visit is 1 before chunking and 4.4 after chunking. The average number of codes per visit is 13.4.

Diagnoses in MIMIC-III are recorded as ICD-9 codes, and we truncate the codes by taking only the first three digits of the ICD-9 codes (first four digits if it's an E code), discarding minor distinctions between codes to obtain broader groupings of codes resulting in 1068 unique codes.

6.4.2 Clinical Language Models

We select two different transformer-based clinical LMs of significantly different model sizes and training corpora in order to expand the settings in which our approach can be tested.

MeDAL-ELECTRA A pretrained ELECTRA-small [47] discriminator further pretrained on MeDAL, a large medical abbreviation disambiguation dataset designed for language model pretraining [169].

Bio-ClinicalBERT A BioBERT-base [101] further pretrained on the MIMIC-III corpus [8]. Note that Bio_ClinicalBERT has more than 8 times the number of parameters as MeDAL_ELECTRA (109M vs. 13.5M).

6.4.3 Multimodality through vocabulary expansion

The set of 1068 truncated ICD codes can be viewed as the vocabulary of diagnoses and can be easily added to the vocabulary of the pretrained LMs through the `add_tokens` method of the corresponding pretrained tokenizer. This method adds the list of ICD codes to the model's vocabulary and returns the number of added tokens, which is subsequently passed to the `resize_token_embeddings` method of the pretrained model to expand the embedding tensors by the appropriate size.

These newly added embeddings are then randomly initialized prior to training. Note that the truncated ICD codes are prepended with the "ICD" string in order to ensure that the codes are not already included in the default vocabulary.

This approach, while simple, is an intuitive and effective way that leverages the existing framework of transformer-based architectures to combine different modalities into a common token space, with their representations learned jointly during end-to-end training.

6.4.4 Graph representation module

We also attach a graph representation module to the last encoder layer of the LM. More specifically, we add the module alongside the self-attention module in the last encoder layer, and the output from the previous (second-to-last) encoder layer is passed concurrently to the self-attention module and to the graph representation module. The outputs from these two parallel modules are then combined through the output module of the attention module prior to being passed to the final feed-forward network, as depicted in Figure 1. The module was added only to the last layer of the model for efficiency.

The implementation of the graph representation module is motivated by [49]. It involves doing a linear projection, calculating the pairwise distance matrix of the projected embeddings, and constructing a graph by converting the pairwise distance matrix into a weighted adjacency matrix A as given by

$$A = T(2 * (1 - \sigma(P))), \tag{6.1}$$

where P is the pairwise distance matrix, σ is the sigmoid function, and T is a threshold function with a hyperparameter t such that

$$T(x) = \begin{cases} x, & \text{if } x \geq t \\ 0, & \text{otherwise} \end{cases} \tag{6.2}$$

Thus the weights α_{ij} in A indicate the strength of connections between tokens at positions i and j in the input sequence, and the non-zero entries in A can be used to construct a graph representation of the input. We use the DGL [162] library to construct the graphs and pass them to a weighted variant of the graph convolutional network [94]

$$H_{l+1} = AH_lW, \tag{6.3}$$

where H_l is the hidden states of the tokens at layer l, A is the weighted adjacency matrix constructed in Equation (6.1), and W is the trainable parameters of the GCN. The weighted GCN layer simply updates the hidden states guided by the structure of the graph based on A in a message-passing framework where a weighted sum of the hidden states of neighboring nodes is used to update the hidden state of each node.

The output of the graph module is then combined with the output of the self-attention mechanism using the output module (i.e. dense layer, layer normalization, and skip connection), and the combined representation continues along the rest of the LM architecture to generate the model predictions.

6.4.5 Experimental Setup

We conduct an ablation study with four settings: LMs with just text, LMs with text and codes, LMs with text and codes with the added graph module, and LMs with text and codes but with the graph module replaced by a simple linear layer with the same number of parameters to assess the benefit of using the graph module. We call these settings text-only, text+codes, full, and full-linear, respectively.

All experiments were run on 6 GTX-1080ti GPUs for 3 epochs with a batch size of 8. The learning rate was tuned on the validation set among a range of values (1e-06, 5e-06, 1e-05, 2e-05, 4e-05) and set to 1e-05 for Bio-ClinicalBERT and 2e-05 for MeDAL-ELECTRA. The threshold value was set to 0.75, and the number of layers for the graph module was set to 1 for Bio-ClinicalBERT and 2 for MeDAL-ELECTRA.

6.5 Results

Model Setting	# Params.	Mortality	Readmission
MeDAL-ELECTRA_full	13.82M	.916	.678
MeDAL-ELECTRA_full-linear	13.82M	.908	.654
MeDAL-ELECTRA_text+codes	13.68M	.907	.646
MeDAL-ELECTRA_text-only	13.55M	.846	.603
Bio-ClinicalBERT_full	110.9M	.918	.676
Bio-ClinicalBERT_full-linear	110.9M	.905	.632
Bio-ClinicalBERT_text+codes	109.7M	.903	.625
Bio-ClinicalBERT_text-only	108.9M	.856	.613

Table 6.1: Ablation study results for the mortality and readmission prediction tasks.

The results of the experiments for the four settings, two model types, and two tasks as described in the previous section are shown in Table 6.1.

Notably, significant performance improvement can be obtained just by incorporating codes into the default LM vocabulary (text-only vs. text+codes), with test accuracies for both tasks and model types increasing by several points (0.846 to 0.907 and 0.603 to 0.646 for MeDAL-ELECTRA and 0.856 to 0.903 and 0.613 to 0.625 for Bio-ClinicalBERT).

Adding the graph module (text+codes vs. full) leads to additional improvements (0.908 to 0.916 and 0.654 to 0.678 for MeDAL-ELECTRA and 0.905 to 0.918 and 0.632 to 0.676 for Bio-ClinicalBERT) much more significantly compared to the full-linear setting, demonstrating that using the graph module does help leverage the latent structure of the input to enhance the learned representation.

The total number of parameters for each setting for both model types are also reported. The vocab expansion and graph module each only adds around 1% to the total number of parameters (0.9% and 1% for MeDAL-ELECTRA and 0.7% and 1.1% for Bio-ClinicalBERT), thus providing efficient ways to add these functionalities to an existing framework.

The test accuracies reported are based on aggregating the predictions for all the chunks for each corresponding visit. More specifically, we take the average of the logits for the chunks under each visit prior to using argmax to obtain the visit-level predicted labels, as further discussed in the next section.

6.6 Discussion

To the best of our knowledge, this paper is the first attempt at directly integrating clinical text and medical codes within the framework of transformer-based LMs that go beyond simple concatenation and late fusion. While the results are promising, there are many avenues for further exploration.

6.6.1 Leveraging knowledge sources

Approaches for jointly learning the representation of text and codes should try to leverage the ontologies or knowledge graphs in which the codes are organized (e.g. ICD-10, RxNorm, UMLS) and incorporate the larger context of the domain knowledge rather than simply treating the set of codes as sequences of tokens. We attempted this by initializing the newly added code embeddings with knowledge graph embeddings [150] trained on the ICD knowledge graph but saw little to no immediate improvement in performance.

However, this line of inquiry warrants a more in-depth investigation because the idea of fully incorporating knowledge graphs and terminologies containing expert knowledge and community consensus is an important one. Joint learning of language and knowledge is an open and challenging area of research [23, 125, 149] that is particularly relevant in the biomedical and clinical domains.

6.6.2 Extending to other modalities

We limited the scope of this study to clinical text and diagnosis codes, but future studies should try to incorporate other modalities such as medications, lab tests, procedures, and general medical concepts.

While our approach offers a straightforward way to extend to more modalities, it would be prudent to engage in some preprocessing for each modality to narrow down the space of added vocabularies (for example, by selecting a subset based on frequency of occurrence or truncating to a higher level in a hierarchy).

Currently, transformer-based LMs are trained with a default vocabulary consisting of around 30k subword units as tokens. As the proportion of newly added vocabulary increases, training might become more difficult and expensive. This leads us naturally to a discussion about one of the main concerns regarding currently available transformer-based clinical LMs: the tokenizer and its default vocabulary.

6.6.3 Tokenizers and vocabularies

Virtually all transformer-based clinical LMs that have been released so far inherit the default tokenizer and vocabulary of the original models (e.g. BERT) used as initialization checkpoints for domain-specific pretraining.

A tokenizer for these LMs has two components: the vocabulary text file with one token per line and the actual algorithms for performing the tokenization that converts the input text into numerical form (along with other relevant functions like adding new tokens or converting tokens to indices and vice versa). Specifically, during domain-specific pretraining, the `vocab.txt` file of the checkpoint model is read in to initialize the domain-specific tokenizer, which is thus identical to the tokenizer for the checkpoint model.

The idea of learning a clinical language-specific vocabulary instead of using the default vocabulary from models pretrained on general English corpora has been the subject of many clinical NLP researchers' curiosity. Given that the vocabulary is usually constructed based on some simple language model trained on the corpus (typically based on unigram or byte-pair encoding), the resulting vocabulary reflects the particular structure of the language that underlies the training corpus. The degree to which the distribution of tokens in the learned vocabulary accurately reflects the language in the domain of interest (e.g. the clinical language) should theoretically be important.

However, this idea has so far remained low-priority for most researchers for two main reasons: the LM would have to be trained from scratch using the newly constructed vocabulary without the benefit of transfer learning, making this endeavor cost prohibitive for most researchers; and the trend of domain-specific pretrained LMs that inherit the vocabulary of the general NLP models has so far worked well enough to mitigate some of the curiosity.

Recently, [102] pretrained a RoBERTa model from scratch along with a specialized clinical vocabulary, providing empirical evidence that learning a domain-specific vocabulary instead of using the default vocabulary can be beneficial.

Looking at specific examples of the shortcomings of the current BERT tokenization further aids the argument: "allergies" is tokenized as "all ##er ##gies", "fentanyl" as "fen ##tan ##yl", "unresponsive" as "un ##res ##pon ##sive", "hypotensive" as "h ##yp ##ote ##ns ##ive", "cardiology" as "card ##iology", and many more. Clinically relevant terms are, more often than not, broken into nonsensical fragments.

While this style of subword tokenization has become popular in NLP due to its reasonable vocabulary size and ability to handle unseen words, we speculate that the fragmentation of words—which become increasingly arbitrary as the domain language deviates further from the original training corpora—acts as a serious impediment to effectively learning the representation of the clinical language.

[25] argue that byte-pair encoding (BPE) is suboptimal for LM pretraining com-

pared to unigram LM tokenization. They show that BPE tokenization (similar to BERT tokenization) results in a larger "dead zone" of tokens whose frequency is much lower than the rest of the vocabulary. This effect would be exacerbated when the tokenizer is used in domains it was not trained in. Further, they demonstrate that unigram LM tokenization produces tokens better aligned with the morphology of the language, which would be important for biomedical NLP since a lot of medical and scientific terms are heavily composed of Greek and Latin roots.

Thus, future work in biomedical and clinical LM pretraining should take into consideration the significance of choices regarding tokenization, vocabulary, and the training corpus.

6.6.4 Limited context window

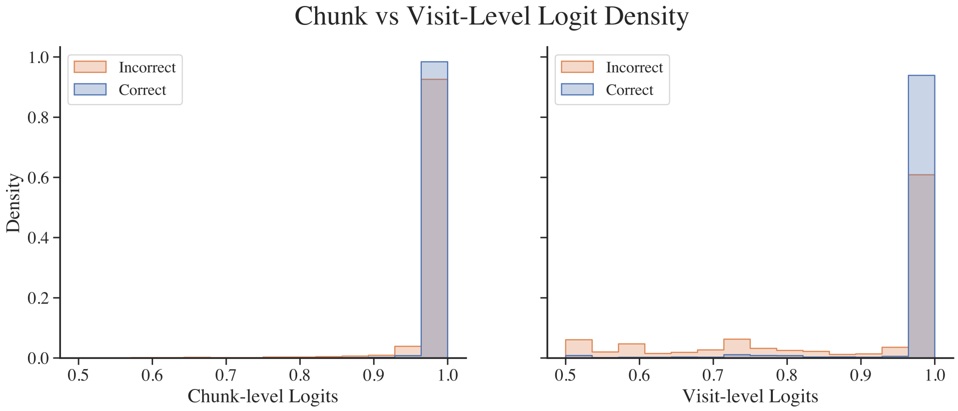


Figure 6.2: Histograms of model outputs for chunks (left) and visits (right), colored by correctness with respect to the visit labels. Counts were transformed into probabilities during plotting for improved visibility of lower logits.

Another major issue with transformer-based clinical LMs is that they also inherit the limited context window sizes from the original pretrained models, which were optimized mostly for sentence-level tasks. Typically, the maximum length these models can handle is 512 tokens. This presents immediate problems for clinical notes, which are often longer than 512 tokens. In order to use these models on clinical notes, we're forced to either truncate the notes at the maximum length and discard the rest, or to add a preprocessing step to break the notes into chunks smaller than the maximum length using some heuristic.

There can also be an optional post-processing step to aggregate the chunk-level predictions (e.g. by averaging the predictions for all chunks pertaining to the same note or the same visit) or an additional trainable module on top of the model to aggregate the chunk representations into a higher-level representation.

In any case, this workaround results in a nontrivial amount of information loss (either from discarding parts of the notes or from breaking down long-range information through chunking) and introduces many potential places for ambiguities and ad hoc decisions that lead to confusion.

For our study, we simply aggregate the chunk-level predictions (logits) into visit-level predictions by taking the average across chunks. This amounts to taking the row-wise mean of the model outputs for each visit prior to taking the argmax for classification.

Figure 6.2 shows side-by-side histograms of chunk-level logits and visit-level logits for mortality prediction, colored by correctness with respect to the visit labels (i.e. expired at the end of visit or not). In the chunk-level histogram, a lot of the incorrect predictions made by the model fall in the higher range of logit values (0.8-1.0), indicating that the model is overconfident. In contrast, the visit-level histogram shows a lot of the incorrect predictions being dispersed along the lower logit values (0.5-0.8) with much less overconfidence in incorrect predictions. Intuitively, this makes sense because predictions made based only on a single chunk will tend to be less informed and accurate than predictions made based on the whole visit. The histograms provide an illustration of the importance of considering visit- versus chunk-level representations, and the accuracies reported in Table 6.1 are based on visit-level predictions.

There are numerous examples in the literature of methods that deal with this issue in various ways. [183] address the problem of limited context windows by incorporating both the position information and time information of chunks across the patients' note series to derive the final patient embeddings. [51] explore several methods to combine chunk-level embeddings to generate encounter-level embeddings for multiple sclerosis consult notes. They try mean, max, and a convolutional neural network (CNN) module [184], with the CNN outperforming the other two, and show that deriving such encounter-level representations is critical to model performance. [87] do something similar by passing the chunk embeddings to a Bi-LSTM model [67] to obtain the final patient representation.

While these approaches do offer useful ways to deal with the limited context window, all of them nonetheless involve chunking the notes to begin with and add many extra steps and decisions to the pipeline.

Another promising direction would be to consider the newer variants of the transformer model with improved efficiency and larger context windows [95, 20, 46]. Such models offer a potentially more elegant way to encode entire clinical notes (or at least larger portions of them) that would reduce the need to create and aggregate chunks of notes.

6.6.5 Graph representation learning

The graph representation module used in this paper is not meant to be a full-fledged graph learning method. Rather, it's intended as a proof of concept for a simple graph-based module that can help enhance the learned representation by incorporating some of the latent structural information in the multimodal inputs.

While one obvious way to improve the module would be to swap out the basic weighted graph convolutional layer with a more sophisticated model like the Graph Transformer [56] or the Graph Attention Network [159], a more compelling albeit challenging avenue would be the graph learning aspect.

Explicit graphs are often not available in clinical data, and manually constructing a graph-based representation of biomedical and clinical data requires considerable expertise and hand-engineering. This makes it difficult to apply existing graph representation learning (GRL) methods, most of which assume that a static, ground-truth graph structure is given.

The open challenge is the learning of graphs or relational structure from data without being given explicit structure (e.g. text, medical codes, images, measurements, etc). Latent graph learning or inference, a fundamental problem in GRL, is the bridge between unstructured/semi-structured data and existing GRL (or even some NLP) methodologies.

There is a new and growing body of literature for graph learning [129], and many of the recent methods use graph neural networks (GNNs). [105] proposed one of the first GNN-based generative models that can generate the entries in a graph adjacency matrix sequentially.

[127] use the Graph Learning Network to simultaneously learn the node embeddings and the edge prediction function based on the node embeddings. Likewise, [61] jointly learn the graph structure of the data and the parameters of a GCN by learning a discrete probability distribution on the adjacency matrix.

Graph Recurrent Attention Network (GRAN) [107] improves upon previous autoregressive graph generation models such as GraphRNN [181], using an attention-based GNN to achieve permutation invariance with respect to node ordering and generating blocks of nodes and edges in linear instead of quadratic autoregressive decision steps.

[49] bring the idea of graph learning into clinical decision support systems and demonstrate that end-to-end graph learning methods that can automatically learn the latent graph structure of patients can offer a powerful alternative to constructing these graphs based on manually defined metrics. The graph representation module implemented in our paper is a variation of their approach.

While the literature on graph learning is still relatively new, recent developments have paved the way for further exploration of this line of research within the biomedical and clinical domains, particularly in a multimodal learning setting in which the relational structure of the data can become increasingly complex.

6.7 Conclusion

The fields of biomedical NLP and medical informatics have experienced rapid progress in recent years. In this paper, we explore the topics of multimodal patient representation learning within the prevalent framework of transformer-based LMs. We also provide detailed discussions about current limitations and directions for future work.

The methods introduced in this paper, particularly the vocabulary expansion, are simple and efficient enough that they could be useful for pretraining a clinical LM given the availability of a training corpus that includes clinical text and corresponding structured codes from the EHR. Such an endeavor, however, would require a considerable amount of investment and coordination on many levels. Nonetheless, we conjecture that the future of biomedical and clinical NLP on the near term will involve newer classes of LMs that can address many of the current limitations and effectively make use of multiple modalities of patient data.

Chapter 7

Conclusion

This dissertation is the end product of the past five years of my time at Yale, roughly the first half of which was spent in a state of confusion and catching up. It wasn't until my third year in 2019 that the field of NLP, deep learning, and my own self all collectively reached a level of bare minimum maturity and preparedness for the conception of this endeavor.

My first real independent project was the 2019 n2c2 shared task over the summer after my qualification into the PhD candidacy. It was a good opportunity to get my hands dirty and carry out the entirety of the life cycle of a research project, from data acquisition and cleaning to the paper write-up and conference presentation. I was lucky enough to, through a mutual friend, find my collaborator Eric, who remains a good friend and a soon-to-be neighbor in Cambridge, MA.

While my contentment with the actual project was partly limited by the disappointing quality of the dataset (which was discussed extensively in the paper, or Chapter 5) and the questionable definition of the task itself, the whole experience did provide me with an array of valuable insights, both technical and political, that informed my future projects and perspectives.

The week Eric and I spent in Washington D.C. during the AMIA Fall Symposium at which we presented our poster for the project was one of the highlights of my graduate school experience and led to our befriending of the competition's winners from IBM, which subsequently led to my internship the following summer of 2020 at IBM Watson Research. The internship, of course, was remote due to the pandemic and proved to be an underwhelming experience.

The chief complaint paper (Chapter 3) was mostly done during late 2019 and early 2020, when the simple application of BERT on all kinds of text data was quickly running its course from being fashionable to being annoyingly uninspired. Even though it was a decent paper, we were lucky to make the cut in terms of its originality and rigor.

The SNOMED-CT knowledge graph embedding paper (Chapter 4) was done over the course of February and March of 2020. I have fond memories of working on this project because it was entirely my idea, and the process was smooth but also filled with interesting twists and insights. Nobody had to hand me a predefined task or a pre-made dataset; I had conceptualized the project and defined its scope based entirely on my readings into the KGE space and my knowledge of biomedical and clinical concept embeddings.

The project was also interesting from the perspective of collaboration. Dan, who worked mostly on pathology problems on the other side of my cubicle and had no business dealing with KGEs, was able to assist me by walking me through the part of a background paper's code that was implemented in R and prompting insights about some of the interesting flaws in existing methodologies. These insights helped guide the experiments and discussions in the paper.

Two other collaborators, Ivana and Carl, who reside on the other side of the pond in Scotland, had written many papers in the KGE literature that I found particularly well-written. The collaboration was formed after my initial chain of emailed questions and their thoughtful replies evolved over the span of weeks into specific discussions about my own project and ultimately their involvement in an official capacity.

The last paper (Chapter 6) materialized over the course of January and February of 2021 as the culmination of two years of foraging through the literature of various adjacent fields and pestering my acquaintances with my earnest hopes of one day writing a paper that involves multimodal learning on clinical data. The premise is simple: given that transformer-based language models have become so dominant in the clinical NLP literature, we should try to think of ways to incorporate multiple modalities of patient data into the widely used framework, which had proven itself to be effective for a wide variety of tasks and data.

Conveniently, the deadline for the 2021 BioNLP workshop at NAACL coincided with the deadline for my dissertation, making it mentally easier for me to merge it into my dissertation as the last chapter and continue to push myself until the finish line.

The paper was very much intentionally named after my dog, Oreo, in a kind of self-gratifying attempt at thumbing my nose in the face of one of many absurd conventions in academic publishing where clever acronyms are manufactured largely for marketing purposes. Not only do all the letters in OREO appear in the title "Multimodal Patient Representation with Transformers" (which is currently a conventionally acceptable way of naming one's work, arguably), but it also signifies something very much real, meaningful, and lovable, running counter to the crushing pressure in academia to do whatever it takes to collect more citations and grant money.

I come away from this whole experience with excitement and anticipation for the future of AI in healthcare. Having recently accepted a job offer from nference, a biotech startup in Cambridge, MA, I feel grateful that I will be able to make very good use of what I've learned during graduate school to build my career.

The global pandemic that brought much death and destruction to the country coincided with the last year of my PhD, taking it in unexpected directions. 2020 was undeniably a gloomy year, but it also led me down a path of reprioritization and reflection that proved to be undeniably rewarding. The various rabbit holes I went down involving Dostoevsky, David Foster Wallace, and Eric Weinstein, the adherence to a constructive and solid routine, the reaffirmation of family values, and the initiative I took to nail down a job prior to months of emotionally demanding thesis work have turned out to be just as important as the actual work of producing the contents for this document.

It's not easy to explain how I became the kind of person who names his last PhD paper after his dog, or who plans on getting a fish tank so he can slap a "This is Water" sticker on it, or who semi-deliberately coordinates the conclusion of Infinite Jest with his thesis defense date, but it is easy, or imperative, to end this long journey on a cryptically sentimental note.

Every love story is a ghost story, and every PhD thesis is a metamorphosis.

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Khushbu Agarwal, Tome Eftimov, Raghavendra Addanki, Sutanay Choudhury, Suzanne Tamang, and Robert Rallo. Snomed2Vec: Random Walk and Poincare Embeddings of a Clinical Knowledge Base for Healthcare Analytics. *arXiv:1907.08650 [cs, stat]*, July 2019. arXiv: 1907.08650.

[3] Khushbu Agarwal, Tome Eftimov, Raghavendra Addanki, Sutanay Choudhury, Suzanne Tamang, and Robert Rallo. Snomed2vec: Random walk and poincaré embeddings of a clinical knowledge base for healthcare analytics. *CoRR*, abs/1907.08650, 2019.

[4] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Mikhail

Galkin, Sahand Sharifzadeh, Asja Fischer, Volker Tresp, and Jens Lehmann. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework, 2020.

[5] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. Pykeen 1.0: A python library for training and evaluating knowledge graph emebddings. *arXiv preprint arXiv:2007.14175*, 2020.

[6] Carl Allen, Ivana Balazevic, and Timothy M. Hospedales. On Understanding Knowledge Graph Representation. *arXiv:1909.11611 [cs, stat]*, September 2019.

[7] Carl Allen, Ivana Balažević, and Timothy Hospedales. Interpreting knowledge graph relation representation from word embeddings, 2021.

[8] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[9] Dominik Aronsky, Diane Kendall, Kathleen Merkley, Brent C. James, and Peter J. Haug. A comprehensive set of coded chief complaints for the emergency department. *Academic Emergency Medicine*, 8(10):980–989, 2001.

[10] Alan Aronson. Effective mapping of biomedical text to the umls metathesaurus: The metamap program. *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, 2001:17–21, 02 2001.

[11] Jimmy Ba, J. Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv*, abs/1607.06450, 2016.

[12] T. Bai, A. K. Chanda, B. Egleston, and S. Vucetic. Ehr phenotyping via jointly embedding medical concepts and words into a unified vector space. *BMC Medical Informatics and Decision Making*, 18, 2018.

[13] T. Bai, A. K. Chanda, B. L. Egleston, and S. Vucetic. Joint learning of representations of medical concepts and words from ehr data. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 764–769, 2017.

[14] Ivana Balažević, Carl Allen, and Timothy M Hospedales. Tucker: Tensor factorization for knowledge graph completion. In *Empirical Methods in Natural Language Processing*, 2019.

[15] Ivana Balažević, Carl Allen, and Timothy Hospedales. Multi-relational poincar\'e graph embeddings. In *Advances in Neural Information Processing Systems*, 2019.

[16] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks.

[17] Andrew L. Beam, Benjamin Kompa, Allen Schmaltz, Inbar Fried, Griffin Weber, Nathan P. Palmer, Xu Shi, Tianxi Cai, and Isaac S. Kohane. Clinical concept embeddings learned from massive sources of multimodal medical data, 2019.

[18] Brett K Beaulieu-Jones, Isaac S Kohane, and Andrew L Beam. Learning contextual hierarchical structure of medical concepts with poincairé embeddings

to clarify phenotypes. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 24:8—17, 2019.

[19] Iz Beltagy, Kyle Lo, and Arman Cohan. 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)*, pages 3615–3620, Hong Kong, China, November 2019. Association for Computational Linguistics.

[20] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

[21] Asma Ben Abacha, Chaitanya Shivade, and Dina Demner-Fushman. 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*, pages 370–379, Florence, Italy, August 2019. Association for Computational Linguistics.

[22] Olivier Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(90001):267D–270, January 2004.

[23] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In Neil D. Lawrence and Mark Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 127–135, La Palma, Canary Islands, 21–23 Apr 2012. PMLR.

[24] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data.

In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc., 2013.

[25] Kaj Bostrom and Greg Durrett. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online, November 2020. Association for Computational Linguistics.

[26] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. 34(4):18–42.

[27] Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. LibKGE - A knowledge graph embedding library for reproducible research. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 165–174, 2020.

[28] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[29] Tiffany J. Callahan, Ignacio J. Tripodi, Harrison Pielke-Lombardo, and

Lawrence E. Hunter. Knowledge-based biomedical data science. *Annual Review of Biomedical Data Science*, 3(1):23–41, 2020.

[30] Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. pages 122–132, 01 2020.

[31] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[32] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914, Online, July 2020. Association for Computational Linguistics.

[33] David Chang, Ivana Balažević, Carl Allen, Daniel Chawla, Cynthia Brandt, and Andrew Taylor. Benchmark and best practices for biomedical knowledge graph embeddings. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 167–176, Online, July 2020. Association for Computational Linguistics.

[34] Wendy W Chapman, Lee M Christensen, Michael M Wagner, Peter J Haug, Oleg Ivanov, John N Dowling, and Robert T Olszewski. Classifying free-text triage chief complaints into syndromic categories with natural language processing. *Artificial intelligence in medicine*, 33(1):31—40, January 2005.

[35] Wendy W. Chapman, John N. Dowling, and Michael M. Wagner. Classification of emergency department chief complaints into 7 syndromes: A retrospective

analysis of 527,228 patients. *Annals of Emergency Medicine*, 46(5):445 – 455, 2005.

[36] Wendy W Chapman, Prakash M Nadkarni, Lynette Hirschman, Leonard W D'Avolio, Guergana K Savova, and Ozlem Uzuner. Overcoming barriers to NLP for clinical text: the role of shared tasks and the need for additional creative solutions. 18(5):540–543.

[37] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 742–751, Red Hook, NY, USA, 2017. Curran Associates Inc.

[38] Wenhu Chen, Yu Su, Xifeng Yan, and William Wang. Kgpt: Knowledge-grounded pre-training for data-to-text generation. *Proceedings of EMNLP 2020*, 2020.

[39] Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M. Hoffman, Wei Xie, Gail L. Rosen, Benjamin J. Lengerich, Johnny Israeli, Jack Lanchantin, Stephen Woloszynek, Anne E. Carpenter, Avanti Shrikumar, Jinbo Xu, Evan M. Cofer, Christopher A. Lavender, Srinivas C. Turaga, Amr M. Alexandari, Zhiyong Lu, David J. Harris, Dave DeCaprio, Yanjun Qi, Anshul Kundaje, Yifan Peng, Laura K. Wiley, Marwin H. S. Segler, Simina M. Boca, S. Joshua Swamidass, Austin Huang, Anthony Gitter, and Casey S. Greene. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, 2018.

[40] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014. Association for Computational Linguistics.

[41] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. Multi-layer Representation Learning for Medical Concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pages 1495–1504, San Francisco, California, USA, 2016. ACM Press.

[42] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[43] Edward Choi, Cao Xiao, Walter Stewart, and Jimeng Sun. MiME: Multilevel medical embedding of electronic health records for predictive healthcare. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4547–4557. Curran Associates, Inc.

[44] Edward Choi, Zhen Xu, Yujia Li, Michael Dusenberry, Gerardo Flores, Emily Xue, and Andrew Dai. Learning the graphical structure of electronic health records with graph convolutional transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):606–613, Apr. 2020.

[45] Youngduck Choi, Chill Yi-I Chiu, and D. Sontag. Learning low-dimensional representations of medical concepts. *AMIA Summits on Translational Science Proceedings*, 2016:41 – 50, 2016.

[46] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers, 2021.

[47] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

[48] Mike Conway, John N. Dowling, and Wendy W. Chapman. Using chief complaints for syndromic surveillance: A review of chief complaint based classifiers in north america. *Journal of Biomedical Informatics*, 46(4):734 – 743, 2013.

[49] Luca Cosmo, Anees Kazi, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael Bronstein. Latent patient network learning for automatic diagnosis. *arXiv preprint arXiv:2003.13620*, 2020.

[50] Hercules Dalianis. *Clinical Text Mining: Secondary Use of Electronic Patient Records*. Springer International Publishing.

[51] Alister D'Costa, Stefan Denkovski, Michal Malyska, Sae Young Moon, Brandon Rufino, Zhen Yang, Taylor Killian, and Marzyeh Ghassemi. Multiple sclerosis severity classification from clinical text. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 7–23, Online, November 2020. Association for Computational Linguistics.

[52] Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi

Tsujii, editors. *Proceedings of the 18th BioNLP Workshop and Shared Task*, Florence, Italy, August 2019. Association for Computational Linguistics.

[53] Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi Tsujii, editors. *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, Online, July 2020. Association for Computational Linguistics.

[54] Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, pages 1811–1818, February 2018.

[55] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[56] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

[57] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.

[58] Keith Feldman, Nicholas Hazekamp, and Nitesh V. Chawla. Mining the clinical narrative: All text are not equal. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 271–280. IEEE.

[59] M. Fey and J. E. Lenssen. Fast graph representation learning with pytorch geometric. *ArXiv*, abs/1903.02428, 2019.

[60] Michele Filannino and Özlem Uzuner. Advancing the state of the art in clinical natural language processing through shared tasks. 27(1):184–192.

[61] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1972–1982. PMLR, 09–15 Jun 2019.

[62] Carol Friedman, Pauline Kra, and Andrey Rzhetsky. Two biomedical sublanguages: a description based on the theories of zellig harris. 35(4):222–235.

[63] Mor Geva, R. Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *ArXiv*, abs/2012.14913, 2020.

[64] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry.

[65] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78 – 94, 2018.

[66] Alex Graves. Generating sequences with recurrent neural networks.

[67] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks : the official journal of the International Neural Network Society*, 18:602–10, 07 2005.

[68] Nathaniel R Greenbaum, Yacine Jernite, Yoni Halpern, Shelley Calder, Larry A Nathanson, David A Sontag, and Steven Horng. Improving documentation of presenting problems in the emergency department using a domain-specific ontology and machine learning-driven user interfaces. *International journal of medical informatics*, 132:103981, December 2019.

[69] Richard T. Griffey, Jesse M. Pines, Heather L. Farley, Michael P. Phelan, Christopher Beach, Jeremiah D. Schuur, and Arjun K. Venkatesh. Chief complaint–based performance measures: A new focus for acute care quality measurement. *Annals of Emergency Medicine*, 65(4):387 – 395, 2015.

[70] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[71] Kelvin Guu, John Miller, and Percy Liang. Traversing knowledge graphs in vector space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 318–327, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[72] Stephanie W. Haas, Debbie Travers, Judith E. Tintinalli, Daniel Pollock, Anna Waller, Edward Barthell, Catharine Burt, Wendy Chapman, Kevin Coonan, Donald Kamens, and James McClay. Toward vocabulary control for chief complaint. *Academic Emergency Medicine*, 15(5):476–482, 2008.

[73] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. 1 2008.

[74] William L. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159.

[75] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[76] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2020.

[77] Frank Heny. Zellig harris, a grammar of english on mathematical principles. new york: Wiley, 1982. pp. xvi+429. 20(1):181–188.

[78] B. Hettige, Yuan-Fang Li, W. Wang, S. Le, and Wray L. Buntine. Medgraph: Structural and temporal representation learning of electronic medical records. In *ECAI*, 2020.

[79] Robert E. Hirschtick. Copy-and-Paste. *JAMA*, 295(20):2335–2336, 05 2006.

[80] Frank L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927.

[81] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. 06(2):107–116.

[82] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. 9(8):1735–1780.

[83] Woo Suk Hong, Adrian Daniel Haimovich, and R. Andrew Taylor. Predicting hospital admission at emergency department triage using machine learning. *PLOS ONE*, 13(7):1–13, 07 2018.

[84] Woo Suk Hong, Adrian Daniel Haimovich, and Richard Andrew Taylor. Predicting 72-hour and 9-day return to the emergency department using machine learning. *JAMIA Open*, 2(3):346–352, 07 2019.

[85] Steven Horng, Nathaniel R. Greenbaum, Larry A. Nathanson, James C McClay, Foster R. Goss, and Jeffrey A. Nielson. Consensus development of a modern ontology of emergency department presenting problems – the hierarchical presenting problem ontology (happy). *bioRxiv*, 2019.

[86] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv:1904.05342*, 2019.

[87] Kexin Huang, Abhishek Singh, Sitong Chen, Edward Moseley, Chih-Ying Deng, Naomi George, and Charolotta Lindvall. Clinical XLNet: Modeling sequential clinical notes and predicting prolonged mechanical ventilation. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 94–100, Online, November 2020. Association for Computational Linguistics.

[88] D. Hüske-Kraus. Text generation in clinical medicine – a review. 42(1):51–60.

[89] Yacine Jernite, Yoni Halpern, Steven Horng, and David Sontag. Predicting chief complaints at triage time in the emergency department. *NeurIPS Workshop on Machine Learning for Clinical Data Analysis and Healthcare*, 2013.

[90] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[91] Seyed Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems 32*, 2018.

[92] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems.

[93] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks.

[94] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[95] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer, 2020.

[96] Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. IMoJIE: Iterative memory-based joint open information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5871–5886, Online, July 2020. Association for Computational Linguistics.

[97] Sotiris Kotitsas, Dimitris Pappas, Ion Androutsopoulos, Ryan McDonald, and Marianna Apidianaki. Embedding Biomedical Ontologies by Jointly Encoding Network Structure and Textual Node Descriptors. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 298–308, Florence, Italy, 2019. Association for Computational Linguistics.

[98] Isotta Landi, Benjamin S Glicksberg, Hao-Chih Lee, Sarah Cherng, Giulia Landi, Matteo Danieletto, Joel T Dudley, Cesare Furlanello, and Riccardo Miotto. Deep representation learning of electronic health records to unlock patient stratification at scale. *NPJ digital medicine*, 3(1):1–11, 2020.

[99] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[100] Dongha Lee, Xiaoqian Jiang, and Hwanjo Yu. Harmonized representation learning on dynamic ehr graphs. *Journal of Biomedical Informatics*, 106:103426, 2020.

[101] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 09 2019.

[102] Patrick Lewis, Myle Ott, Jingfei Du, and Veselin Stoyanov. Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 146–157, Online, November 2020. Association for Computational Linguistics.

[103] Yikuan Li, S. Rao, J. R. A. Solares, A. Hassaine, R. Ramakrishnan, D. Canoy, Y. Zhu, K. Rahimi, and G. Salimi-Khorshidi. Behrt: Transformer for electronic health records. *Scientific Reports*, 10, 2020.

[104] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. *CoRR*, abs/1904.12787, 2019.

[105] Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs, 2018.

[106] Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. Gated graph sequence neural networks. In *Proceedings of ICLR'16*, April 2016.

[107] Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Charlie Nash, William L. Hamilton, David Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. In *NeurIPS*, 2019.

[108] Feifan Liu, Chunhua Weng, and Hong Yu. Advancing clinical research through natural language processing on electronic health records: Traditional machine learning meets deep learning. In Rachel L. Richesson and James E. Andrews, editors, *Clinical Research Informatics*, Health Informatics, pages 357–378. Springer International Publishing.

[109] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph.

Proceedings of the AAAI Conference on Artificial Intelligence, 34(03):2901–2908, Apr. 2020.

[110] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy, July 2019. Association for Computational Linguistics.

[111] Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019.

[112] Christian Merkwirth and Thomas Lengauer. Automatic generation of complementary descriptors with molecular graph networks. *Journal of chemical information and modeling*, 45:1159–68, 09 2005.

[113] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[114] Martin Mockel, Julia Searle, Reinhold Muller, Anna Slagman, Harald Storchmann, Philipp Oestereich, Werner Wyrwich, Angela Ale-Abaei, Joern O Vollert, Matthias Koch, and Rajan Somasundaram. Chief complaints in medical emergencies: do they relate to underlying disease and outcome? the charité emergency medicine study (charitem). *European journal of emergency medicine : official journal of the European Society for Emergency Medicine*, 20(2):103—108, April 2013.

[115] Hani Mowafi, Daniel Dworkis, Mark Bisanzo, Bhakti Hansoti, Phil Seidenberg,

Ziad Obermeyer, Mark Hauswald, and Teri A. Reynolds. Making recording and analysis of chief complaint a priority for global emergency care research in low-income countries. *Academic Emergency Medicine*, 20(12):1241–1245, 2013.

[116] Karthik Murugadoss, Ajit Rajasekharan, Bradley Malin, Vineet Agarwal, Sairam Bade, Jeff R. Anderson, Jason L. Ross, William A. Faubion, John D. Halamka, Venky Soundararajan, and Sankar Ardhanari. Building a best-in-class de-identification tool for electronic medical records through ensemble learning. *medRxiv*, 2020.

[117] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[118] Stuart Nelson, Kelly Zeng, John Kilbourne, Tammy Powell, and Robin Moore. Normalized names for clinical drugs: RxNorm at 6 years. *JAMIA*, 18(4):441–448, 4 2011.

[119] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegal. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*, pages 809–816, 2011.

[120] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS-W*.

[121] Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pages 58–65, 2019.

[122] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

[123] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[124] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations.

[125] Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China, November 2019. Association for Computational Linguistics.

[126] Matthew E. Peters, Mark Neumann, Robert L Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In *EMNLP*, 2019.

[127] Darwin Danilo Saire Pilco and Adín Ramírez Rivera. Graph learning network: A structure learning algorithm, 2019.

[128] Tom J. Pollard, Alistair E. W. Johnson, J. Raffa, L. Celi, R. Mark, and O. Badawi.

The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific Data*, 5, 2018.

[129] Lishan Qiao, Limei Zhang, Songcan Chen, and Dinggang Shen. Data-driven graph construction and graph learning: A review. *Neurocomputing*, 312:336–351, 2018.

[130] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[131] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu Galtier, Bennett Landman, Klaus Maier-Hein, Sébastien Ourselin, Micah Sheller, Ronald Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and Manuel Jorge Cardoso. The future of digital health with federated learning. *npj Digital Medicine*, 3, 12 2020.

[132] Kristin L. Rising, Timothy W. Victor, Judd E. Hollander, and Brendan G. Carr. Patient returns to the emergency department: The time-to-return curve. *Academic Emergency Medicine*, 21(8):864–871, 2014.

[133] Emma Rocheteau, Catherine Tong, Petar Veličković, Nicholas Lane, and Pietro Liò. Predicting patient outcomes with graph representation learning, 2021.

[134] Alexey Romanov and Chaitanya Shivade. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1586–1596. Association for Computational Linguistics.

[135] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and

validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987.

[136] Sebastian Ruder. Neural transfer learning for natural language processing.

[137] Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You {can} teach an old dog new tricks! on training knowledge graph embeddings. In *International Conference on Learning Representations*, 2020.

[138] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning Internal Representations by Error Propagation*, page 318–362. MIT Press, Cambridge, MA, USA, 1986.

[139] Anna Rumshisky, Kirk Roberts, Steven Bethard, and Tristan Naumann, editors. *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, Online, November 2020. Association for Computational Linguistics.

[140] Tara Safavi, Danai Koutra, and Edgar Meij. Evaluating the Calibration of Knowledge Graph Embeddings for Trustworthy Link Prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8308–8321, Online, November 2020. Association for Computational Linguistics.

[141] Guergana Savova, Sameer Pradhan, Martha Palmer, Will Styler, Wendy Chapman, and Noémie Elhadad. Annotating the clinical text – MiPACQ, ShARe, SHARPn and THYME corpora. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 1357–1378. Springer Netherlands.

[142] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[143] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *Trans. Neur. Netw.*, 20(1):61–80, January 2009.

[144] Benjamin Shickel, Patrick James Tighe, Azra Bihorac, and Parisa Rashidi. Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. *IEEE journal of biomedical and health informatics*, 22(5):1589–1604, 2017.

[145] Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, 26(11):1297–1304, 07 2019.

[146] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[147] Ethan Steinberg, Ken Jung, Jason A Fries, Conor K Corbin, Stephen R Pfohl, and Nigam H Shah. Language models are an effective representation learning technique for electronic health record data. *Journal of Biomedical Informatics*, 113:103637, 2021.

[148] Lina Sulieman, David Gilmore, Christi French, Robert M. Cronin, Gretchen Purcell Jackson, Matthew Russell, and Daniel Fabbri. Classifying patient portal messages using convolutional neural networks. *Journal of Biomedical Informatics*, 74:59 – 70, 2017.

[149] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 2.0: A continual pre-training framework for language

understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8968–8975, Apr. 2020.

[150] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019.

[151] Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web*, pages 287–297. International World Wide Web Conferences Steering Committee, 2016.

[152] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey, 2020.

[153] The Gene Ontology Consortium. The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Research*, 47(D1):D330–D338, 11 2018.

[154] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning (ICML)*, volume 48, pages 2071–2080, 2016.

[155] Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. 18(5):552–556.

[156] Ilya Valmianski, Caleb Goodwin, Ian M. Finn, Naqi Khan, and Daniel S. Zisook. Evaluating robustness of language models for chief complaint extraction from patient-generated text. *CoRR*, abs/1911.06915, 2019.

[157] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[158] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[159] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks.

[160] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[161] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. 2019. In the Proceedings of ICLR.

[162] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

[163] Quan Wang, Pingping Huang, Haifeng Wang, Songtai Dai, Wenbin Jiang, Jing Liu, Yajuan Lyu, and Hua Wu. Coke: Contextualized knowledge graph embedding. *arXiv:1911.02168*, 2019.

[164] Yanshan Wang, Naveed Afzal, Sunyang Fu, Liwei Wang, Feichen Shen, Majid

Rastegar-Mojarad, and Hongfang Liu. Medsts: a resource for clinical semantic textual similarity. *Language Resources and Evaluation*, 54(1):57–72, March 2020. Funding Information: This work was made possible by the National Institute of Health (NIH) grants R01LM011934, R01GM102282, R01EB19403, R01LM11829 and U01TR002062.

[165] Yanshan Wang, Sunyang Fu, Feichen Shen, Sam Henry, Ozlem Uzuner, and Hongfang Liu. The 2019 n2c2/ohnlp track on clinical semantic textual similarity: Overview. *JMIR Med Inform*, 8(11):e23375, Nov 2020.

[166] Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, and Hongfang Liu. Clinical information extraction applications: A literature review. 77:34–49.

[167] Yuxia Wang, Fei Liu, Karin Verspoor, and Timothy Baldwin. Evaluating the utility of model configurations and data augmentation on clinical semantic textual similarity. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 105–111, Online, July 2020. Association for Computational Linguistics.

[168] Marinka Zitnik Yuxiao Dong Hongyu Ren Bowen Liu Michele Catasta Jure Leskovec Weihua Hu, Matthias Fey. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

[169] Zhi Wen, Xing Han Lu, and Siva Reddy. MeDAL: Medical abbreviation disambiguation dataset for natural language understanding pretraining. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 130–135, Online, November 2020. Association for Computational Linguistics.

[170] Wei-Hung Weng and Peter Szolovits. Representation learning for electronic health records. *arXiv preprint arXiv:1909.09248*, 2019.

[171] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[172] Stephen Wu, Kirk Roberts, Surabhi Datta, Jingcheng Du, Zongcheng Ji, Yuqi Si, Sarvesh Soni, Qiong Wang, Qiang Wei, Yang Xiang, Bo Zhao, and Hua Xu. Deep learning in clinical natural language processing: a methodical review. *Journal of the American Medical Informatics Association*, 27(3):457–470, 12 2019.

[173] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks.

[174] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?

[175] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5449–5458. PMLR, 2018.

[176] Yichong Xu, Xiaodong Liu, Chunyuan Li, Hoifung Poon, and Jianfeng Gao. DoubleTransfer at MEDIQA 2019: Multi-source transfer learning for natural language understanding in the medical domain. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 399–405, Florence, Italy, August 2019. Association for Computational Linguistics.

[177] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online, November 2020. Association for Computational Linguistics.

[178] Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, May 2015.

[179] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[180] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification.

[181] Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models, 2018.

[182] Xianlong Zeng, Yunyi Feng, Soheil Moosavinasab, Deborah Lin, Simon Lin, and

Chang Liu. Multilevel self-attention model and its use on medical risk prediction. In *Pac Symp Biocomput*. World Scientific, 2020.

[183] Dongyu Zhang, Jidapa Thadajarassiri, Cansu Sen, and Elke Rundensteiner. Time-aware transformer-based network for clinical notes series prediction. In Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 5th Machine Learning for Healthcare Conference*, volume 126 of *Proceedings of Machine Learning Research*, pages 566–588, Virtual, 07–08 Aug 2020. PMLR.

[184] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification, 2016.

[185] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics.

[186] Ziwei Zhang, Peng Cui, and Wenwu Zhu. Deep learning on graphs: A survey.

[187] Da Zheng, Xiang Song, Chao Ma, Zeyuan Tan, Zihao Ye, Jin Dong, Hao Xiong, Zheng Zhang, and George Karypis. Dgl-ke: Training knowledge graph embeddings at scale. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 739–748, New York, NY, USA, 2020. Association for Computing Machinery.

[188] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications.

[189] Henghui Zhu, Ioannis Ch. Paschalidis, and Amir Tahmasebi. Clinical concept extraction with contextual word embedding. *CoRR*, abs/1810.10566, 2018.

[190] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[191] Zhaocheng Zhu, Shizhen Xu, Meng Qu, and Jian Tang. Graphvite: A high-performance cpu-gpu hybrid system for node embedding. In *The World Wide Web Conference*, pages 2494–2504. ACM, 2019.

[192] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 06 2018.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346 USA