FUSING MULTIMODAL KNOWLEDGE IN LANGUAGE MODELS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Michihiro Yasunaga
April 2024

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Percy Liang, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Jure Leskovec, Co-Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Christopher Manning

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format.

Abstract

Language models, such as GPT-4, have the capability to generate textual responses to user queries. They are used across various tasks, including question answering, translation, summarization, and personal assistance. However, to create more versatile AI assistants, these models need to handle more diverse and complex tasks involving domain or visual knowledge, such as answering medical questions and explaining or generating images. This necessity motivates the development of models that can access and leverage diverse knowledge sources beyond text, such as databases and images.

In this thesis, we aim to develop language models capable of using multimodal knowledge, encompassing text, knowledge graphs, and images, to address various user queries. Text provides broad and contextually rich knowledge, knowledge graphs often supply structured domain knowledge, and images facilitate various visual applications.

This thesis consists of five chapters. The first chapter introduces methods for language models to efficiently learn knowledge from textual data. Specifically, we train language models on a sequence of multiple related documents, encouraging them to learn and reason about knowledge with long-range dependencies. This approach yields strong performance on complex long-context and multi-step reasoning tasks. In the second chapter, we introduce methods that enable language models to harness knowledge graph information. Specifically, we develop a new model architecture, a hybrid of language models and graph neural networks, along with a training objective that fuses text and knowledge graph representations. This method demonstrates strong performance on tasks involving domain knowledge, such as medical question answering. In the third chapter, to empower language models to use and generate visual content alongside textual information, we design unified multimodal models capable of encoding, retrieving, and decoding interleaved sequences of text and images. The model employs a retriever to fetch textual or visual knowledge and integrates it into a multimodal Transformer that encodes and decodes both text and images using token representations. Finally, in the forth and fifth chapters, we demonstrate the application of textual, structured, and visual knowledge fusion techniques to solve practical healthcare tasks, including clinical trial outcome prediction and multimodal medical question answering.

In summary, this thesis builds models capable of comprehending and generating multimodal content, spanning text, knowledge graphs, and images.

Acknowledgments

My PhD would not have been possible without the support of my mentors, collaborators, friends, and family.

I want to first express my gratitude to my PhD advisors, Percy Liang and Jure Leskovec, who played a significant role in shaping my doctoral journey and this dissertation.

Percy is not only an exceptional researcher, but also a dedicated mentor who truly cares about his students. He guided me patiently as I explored different research interests, including language, knowledge, reasoning, and multimodality. Percy gave me the opportunities to delve into these areas and grow as a researcher. His academic excellence, especially his emphasis on clarity, rigor, and big picture, has shaped how I approach research. Our deep discussions and the incredibly thorough and insightful comments Percy provided me on every paper draft have pushed me to grow and enhance my research significantly. I feel extremely fortunate to be advised by him.

Jure is an exceptional researcher, thought leader, and mentor. Throughout my PhD, I learned invaluable lessons from Jure—from how to develop and frame impactful ideas to how to write papers and make presentations in a way that excites the audience. I always aimed to be well-prepared for our meetings, but Jure's high standards and fresh perspectives consistently enabled me to elevate my work. Jure also provided me with ample opportunities to collaborate with diverse researchers, including medical researchers and industry researchers, which broadened the scope and depth of my research. I am truly grateful for his mentorship.

I would also like to express my gratitude to my dissertation committee for their guidance. I am thankful to Chris Manning for serving as my first-year rotation advisor and for providing valuable advice on my dissertation research and proposal. I gained substantial insights from his profound expertise in NLP. I am grateful to Tatsu Hashimoto for his guidance on my dissertation research, as well as for all our research discussions and collaborations. I want to thank Chris Potts for agreeing to be the oral committee chair and for his excellent suggestions during the defense.

I am grateful for my mentors from my undergraduate studies, who continue to inspire me and serve as my role models during my PhD studies. Dragomir Radev was my first research mentor, who ignited my interest in NLP and AI research. He taught me the importance of having confidence in my research, presenting and disseminating my ideas, and building my network in the research

community. John Lafferty imparted to me the fundamentals needed to design rigorous machine learning models and algorithms. Rui Zhang and Tao Yu were PhD students who mentored me during my undergraduate studies, and we spent a lot of time collaborating on NLP research. I learned the fun and importance of research collaboration from them.

Over the past five years, I have had the privilege of working with many amazing colleagues and collaborators. I am grateful to my collaborators, including Hongyu Ren, Tony Lee, Antoine Bosselut, Maria Brbic, Pang Wei Koh, Michael Moor, Hamed Nilforoshan, Yanan Wang, Rishi Bommasani, Shiori Sagawa, Joon Sung Park, Yifan Mai, Qian Huang, Michael Xie, Ananya Kumar, Nelson Liu, Sidd Karamcheti, Lisa Li, John Hewitt, Mina Lee, Dimitris Tsipras, Niladri Chatterji, Weihua Hu, Shirley Wu, Kaidi Cao, Xikun Zhang, Yuhui Zhang, Jeff HaoChen, Yusuf Roohani, Yanay Rosen, Camilo Ruiz, Kexin Huang, Chenlin Meng, Jun-Yan Zhu, Jiajun Wu, Fei-Fei Li, Prabhat Agarwal, Josselin Somerville, Chi Heem Wong, and Sharmila Reddy Nangi. I am also grateful to Rok Sosic for his research and career advice. I feel very fortunate to have worked with the brightest minds to develop my research work. I want to particularly thank Hongyu and Tony. Hongyu and I had a close collaboration on the projects developing knowledge-augmented language models. Tony and I had a close collaboration on the projects evaluating language models, text-to-image models, and vision-language models.

I extend my gratitude to all the members of the P-Lambda group (a research lab led by Percy), the SNAP group (a research lab led by Jure), and the Stanford Natural Language Processing group. I have had the privilege of meeting and interacting with an incredible set of people with diverse research backgrounds, spanning core machine learning and NLP to theory, biology, medicine, and social sciences. The group members introduced fascinating research topics that I might not have encountered otherwise. Being part of these groups during my PhD was truly inspiring.

The research presented in this dissertation would not have been possible without the support of the following organizations: Funai Foundation Fellowship, Microsoft Research Fellowship, Defense Advanced Research Projects Agency (DARPA), and the National Science Foundation (NSF).

My research experience was also shaped by the internships I completed during my PhD years. I would like to express my gratitude to Scott Yih, Luke Zettlemoyer, Armen Aghajanyan, Mike Lewis, and Rich James for their support during my internship at Meta; and Denny Zhou, Xinyun Chen, Ed H. Chi, Yujia Li, and Panupong Pasupat for their support during my internship at Google Brain/DeepMind. They made my internships feel like a second home. I am also thankful for the friends I met during my internships, with whom I had invaluable conversations: Weijia Shi, Sida Wang, Chunting Zhou, Sewon Min, Asli Celikyilmaz, Victor Zhong, Akari Asai, Yizhong Wang, Yushi Hu, Daniel Fried, Hanjun Dai, Eric Wallace, and Alisa Liu.

I also want to thank my wonderful friends in the research community: Irene Li, Jungo Kasai, Yutaro Yamada, and Alexander Fabbri, for spending time together and engaging in invaluable conversations.

I am deeply grateful to my family. My parents, my sister, and my partner have always believed in my abilities more than anyone else. They have consistently supported me in pursuing my dreams and enabled me to achieve far more than I could have ever imagined. They have also been a constant source of fresh perspectives, inspirations, and engaging conversations, and have enriched my academic journey. I extend my heartfelt thanks to my parents, my sister, and my partner for their love and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Language models, such as GPT-4 (Brown et al., 2020), are machine learning models designed to comprehend and generate textual content. These models are typically large neural networks trained by exposing them to extensive textual datasets and teaching them to predict next words (Radford et al., 2019). As a result of this training, the models learn to take input x, a sequence of words, and generate corresponding responses y. This fundamental capability enables language models to be applied across a range of language-related tasks, including text completion, question answering, translation, and summarization (Brown et al., 2020; Qin et al., 2023; Bommasani et al., 2021).

Question answering serves as a fundamental function in many products and services, such as search engines, personal assistants, and automated help systems (Pieraccini et al., 1991; Hendrix et al., 1978; Raymond and Riccardi, 2007; Weston et al., 2015; McCann et al., 2018). In practice, examples of user queries span a wide range of topics and complexity, requiring different forms of knowledge. For instance, questions like "Which countries in Europe have not hosted World Cup?" necessitate general encyclopedia knowledge, while queries like "What are FDA-approved drugs to treat breast cancer?" demand specialized domain knowledge in medicine. Inquiries like "Please depict what an Armenian church looks like" extend beyond text-based knowledge, involving a requirement for visual information.

However, for language models to proficiently perform such tasks, access to diverse knowledge sources is imperative. Knowledge, in this context, refers to pieces of information useful for performing a task. For example, without sufficient domain knowledge, such as in medicine, models might produce inaccurate answers (e.g., drug names) to medical queries, leading to potentially serious consequences. Moreover, as human communication and perception are inherently multimodal, with a predominant reliance on visual processing, the absence of visual knowledge significantly limits the applications of models.

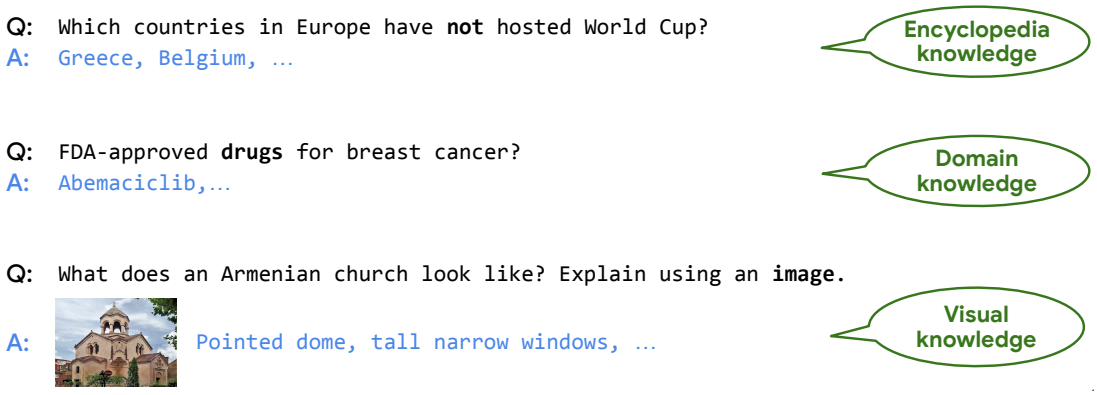


Figure 1.1: Examples of user queries requiring various types of knowledge, such as encyclopedia knowledge, domain knowledge, and visual knowledge.

Therefore, to develop a truly versatile system capable of responding to various user requests, it is necessary to develop language models capable of accessing and utilizing diverse sources of knowledge. This is the central goal of this thesis. The primary technical challenge lies in the fact that knowledge exists in diverse formats and modalities, including text, structured knowledge bases, and images, and all of them offer complementary information:

- For instance, textual knowledge encompasses general-domain documents such as Wikipedia and web texts, as well as domain-specific documents like PubMed and electronic health records (Gao et al., 2020). Text is one of the most common media for communicating and elaborating information, offering broad and context-rich knowledge.

- Structured knowledge, such as knowledge graphs, is another common source of information. A knowledge graph (KG) represents entities as nodes and their relations as edges (e.g., (`Paris`, `in`, `France`)). This includes general domain knowledge graphs like Freebase (Bollacker et al., 2008), Wikidata (Vrandečić and Krötzsch, 2014), and ConceptNet (Speer et al., 2017), as well as domain-specialized (such as medical) knowledge graphs like UMLS (Bodenreider, 2004) and DrugBank (Wishart et al., 2018). Some knowledge graphs are constructed from raw text for better information organization (Speer et al., 2017), and some are manually curated (Fabregat et al., 2018), covering information that may not be ready available in text. Knowledge graphs are useful in providing structured information valuable for complex query answering (Ren et al., 2020) and domain expertise, as professionals in specialized fields such as biology, medicine, and finance often curate domain knowledge in the form of knowledge graphs (Hewett et al., 2002).

- Visual knowledge introduces another dimension of information. Our world inherently involves

multimodal information, particularly visual and language information. In fact, visuals convey the most information in human communication and perception and one image can be worth a thousand words (Burmark, 2002; Standard, 1911). Consequently, language models are increasingly expected to process and generate content incorporating visual elements. Incorporating visuals can enable various multimodal applications, such as generating images or videos from textual descriptions and answering questions or generating text about images, which are becoming more prevalent (Rombach et al., 2022; OpenAI, 2023).

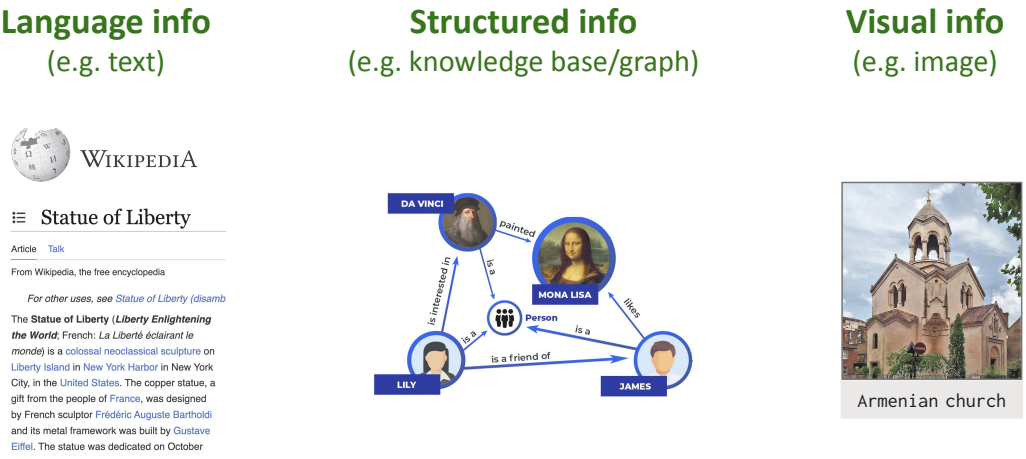


Figure 1.2: Knowledge is available in diverse formats, such as text, structured knowledge graphs, and images.

1.2 Thesis Outline

Towards building language models capable of using diverse and multimodal knowledge, this thesis first develops methods for fusing (I) textual knowledge, (II) structured knowledge, and (III) visual knowledge into language models to perform diverse tasks (Part I). We then present practical applications of these methods in medical scenarios, building clinical trial and question answering systems (Part II).

Chapter 2: Background. Before delving into the main content, we review the fundamental concepts and prior research on knowledge representation and language models.

Part I: Methodologies

Chapter 3: Fusing textual knowledge in models. Textual data provides broad and contextually-rich knowledge. We introduce methods for language models to efficiently learn knowledge from textual data. Specifically, we train language models on a sequence of multiple related documents instead of conventional approaches that rely on single documents. This approach encourages models to learn and reason about complex knowledge that has long-range dependencies. Our training method yields strong performance on various complex NLP tasks, especially long-context and multi-step reasoning tasks.

Chapter 4: Fusing structured knowledge in models. Structured knowledge graphs offer additional information and domain-specific knowledge to complement text. We introduce methods that enable language models to harness knowledge graph information. Specifically, we propose a novel model architecture, a hybrid of language models and graph neural networks, along with a training objective that fuses text and knowledge graph representations. This method outperforms existing language models on various NLP tasks, especially on those involving domain knowledge like medical question answering.

Chapter 5: Fusing visual knowledge in models. Finally, to enable language models to use and generate visual content alongside textual information, we design unified multimodal language models capable of encoding, retrieving, and decoding interleaved sequences of text and images. The model employs a retriever to fetch textual or visual knowledge and integrates it into a multimodal Transformer that encodes and decodes both text and images using token representations. We demonstrate that this knowledge retrieval technique enhances the accuracy of visual and textual generation over existing vision-language models.

Part II: Applications

Chapter 6: Clinical trials. Clinical trials suffer from inefficiency and high costs: in the US, each trial can cost over 10 million dollars and 5 years on average, but 80% of trials fail due to an inability to show efficacy and safety (Ali et al., 2020; Liu et al., 2021b). To address this, we apply the technique of textual and structured knowledge fusion (Chapter 3 and 4) to build a model for predicting the safety and efficacy of a clinical trial using trial documents and clinical knowledge graphs. We show that the model, trained on historical clinical trial data, can accurately predict outcomes for new, previously unseen trials. This suggests the potential to reduce costs and enhance safety and efficacy in future clinical trials.

Chapter 7: Multimodal medical question answering. The increased demand for healthcare motivates the development of fast and accurate interfaces for healthcare providers and patients, such as medical question answering systems (Kirch and Petelle, 2017; Davenport and Kalakota, 2019).

Given the diverse nature of medical data, encompassing text and images such as X-rays, we apply the technique of textual and visual knowledge fusion (Chapter 5) to build a medical QA system that can handle multimodal content. Our system demonstrates a 20% improvement in clinical usefulness compared to prior medical QA systems, as evaluated by clinical experts.

Chapter 8: Conclusion. We conclude and discuss future research directions in multimodal models.

1.3 Contributions

The contributions of this thesis are summarized as follows.

- We were among the first to develop the technique to train language models on multi-document inputs, enabling the learning of broader and more complex knowledge from textual data (Yasunaga et al., 2022b). We show that this technique enhances long-context and multi-step reasoning performance of language models. This result has served as inspiration for subsequent research in the field, leading to the adoption of retrieval-augmented training and multi-document training in larger-scale language model training and their broader applications in biomedicine and healthcare (Shi et al., 2023a; Frisoni et al., 2022).

- We pioneered the research in enabling language models to use structured knowledge graphs. Specifically, we designed an expressive model architecture to fuse knowledge graphs with text (Yasunaga et al., 2021) and training objectives that facilitate joint reasoning across these elements (Yasunaga et al., 2022a). In essence, our research shows that knowledge graphs can offer complementary information to textual data. This insight has inspired subsequent works in the field, which leverage the strengths of both textual data and knowledge graphs across various applications (Sun et al., 2022; Wang et al., 2022a;b).

- We developed the first multimodal language model capable of retrieving and generating interleaved text and images (Yasunaga et al., 2023). Our unified model architecture is designed to retrieve, fuse, and generate textual and visual elements using token representations. We demonstrate that this model not only enhances generation accuracy but also enables novel multimodal in-context learning and prompting capabilities. Our model is inspiring subsequent research in the field, such as extending and scaling the unified model architecture to include additional modalities like speech (Aghajanyan et al., 2023) and to perform a broader range of downstream tasks (Yu et al., 2023).

- We applied the techniques of textual, structured, and visual knowledge fusion to develop practical systems for healthcare, such as accurate clinical trial outcome prediction (Brbic et al.,

2024) and multimodal medical question answering (Moor et al., 2023b). Through evaluations conducted by medical researchers and clinical professionals, our models yield clinically valuable predictions for various aspects of clinical trials, such as drug safety and efficacy, as well as drug repurposing. Additionally, our systems provide clinically useful responses for medical question answering and chatbots, aiding in tasks such as diagnosis from X-ray images.

1.4 Bibliographic Remarks

The research presented in this thesis is based on the following publications and manuscripts.

- Chapter 3:
  - Michihiro Yasunaga, Jure Leskovec*, Percy Liang*. LinkBERT: Pretraining Language Models with Document Links. Association for Computational Linguistics (ACL), 2022.
- Chapter 4:
  - Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D. Manning, Percy Liang*, Jure Leskovec*. DRAGON: Deep Bidirectional Language-Knowledge Graph Pretraining. Neural Information Processing Systems (NeurIPS), 2022.
- Chapter 5:
  - Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Rich James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, Wen-tau Yih. Retrieval-Augmented Multimodal Language Modeling. International Conference on Machine Learning (ICML), 2023.
- Chapter 6:
  - Maria Brbić*, Michihiro Yasunaga*, Prabhat Agarwal*, and Jure Leskovec. Predicting drug outcome of population via clinical knowledge graph. 2024.
- Chapter 7:
  - Michael Moor, Qian Huang, Shirley Wu, Michihiro Yasunaga, Cyril Zakka, Yash Dalmia, Eduardo Pontes Reis, Pranav Rajpurkar, Jure Leskovec. Med-Flamingo: a Multimodal Medical Few-shot Learner. Machine Learning for Health (ML4H), 2023.

Chapter 2

Background

To establish the background for this thesis, we first outline the fundamental concepts about knowledge representations (§2.1). Subsequently, we delve into the historical context surrounding research in knowledge and AI and discuss the positioning of this thesis within it (§2.2).

2.1 Preliminaries

In the context of this thesis, **knowledge** refers to the pieces of information useful for performing a task such as question answering, and reasoning refers to the process of using knowledge to derive answers, which may involve a series of steps (Levesque, 1986). Knowledge comprises two layers: the source data, which consists of raw information (e.g., text, images, graphs), and the model, which processes the raw data into a more accessible form of knowledge that is easier to query (e.g., language model encoder, image encoder, graph encoder).

The source data can take various formats and modalities, including text (§2.1.1), images (§2.1.3), and knowledge bases (§2.1.2). Different data formats may exhibit varying degrees of organization; for example, knowledge bases tend to be more structured and distilled than raw text and images. Moreover, different data formats cover different aspects: text offers broad coverage and contextual richness, knowledge bases provide structured information where concepts are cleanly connected, and images offer a visual dimension that cannot be conveyed through text alone.

Traditionally, different models have been developed to process these three types of data sources: language models like Transformers for text (§2.1.1), convolutional neural networks for images (§2.1.3), and graph neural networks for knowledge graphs (§2.1.2). This thesis aims to develop unified models capable of leveraging these diverse formats of knowledge sources to perform various tasks.

2.1.1 Textual knowledge

Source data

Text, or a document, is a sequence of tokens, typically words. The strength of text as a knowledge source lies in its breadth and contextual richness. Since language serves as a common medium of communication, vast amounts of text are readily available, spanning various sources such as the web and books, which makes it easy to amass large datasets with minimal human labor. Consequently, text data can encompass a wide range of knowledge, spanning various domains from general sources like Wikipedia to specialized ones such as law (e.g., USPTO), science (e.g., arXiv), medicine (e.g., PubMed), and source code repositories (e.g., GitHub) (Gao et al., 2020). Text also facilitates detailed descriptions, explanations, and narratives, providing contextually rich and nuanced information (e.g., instead of just stating "Ibuprofen can treat fever," it can provide additional contexts such as "but it may not be suitable for individuals with stomach ulcers due to potential side effects").

Model

A language model is a machine learning model designed to comprehend and generate sequences of tokens (Bengio et al., 2000). It holds the ability to process raw textual information into a more accessible form of knowledge that is easier to query (Petroni et al., 2019). For instance, a language model pretrained on Wikipedia articles can be prompted with the input, "The author of Harry Potter is __", and it will be able to respond with "JK Rowling".

A language model typically has two types, encoder and decoder:

- An encoder language model, such as BERT (Devlin et al., 2019), processes in a sequence of tokens $X = (x_1, x_2, ..., x_n)$ and generates a contextualized vector representation for every token, $(\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n)$. For example, the Transformer (Vaswani et al., 2017) is a commonly used model architecture. It also incorporates a head, like a linear layer, which uses these vector representations to perform specific tasks on the tokens or the entire sequence, such as text classification or multiple-choice question answering. An encoder language model is typically pre-trained by the masked token prediction task, where it learns to predict masked tokens in the input, and is then fine-tuned on downstream tasks (Devlin et al., 2019).

- A decoder language model, such as GPT-2 and GPT-3 (Radford et al., 2019; Brown et al., 2020), similarly takes in a textual input and generates contextualized representations. However, its head then uses these representations to predict the next token and generate text autoregressively. This can be used for many tasks that can be cast as text generation, such as question answering, summarization, and translation (Bommasani et al., 2021; Liang et al., 2022; Qin et al., 2023). A decoder language model is typically trained by the next token prediction task.

In Chapter 3, we will present our methods to efficiently train language models given a set of documents as training data. We will discuss additional related works on language model training and textual knowledge in §3.6.

2.1.2 Structured knowledge

Source data

A structured knowledge source has a defined schema, specifying the types of objects and their relationships within the knowledge source. Common examples include databases and knowledge bases (Bollacker et al., 2008; Vrandečić and Krötzsch, 2014). A knowledge base consists of sentences stating facts or rules. In modern triplet-based knowledge bases, each sentence consists of a head entity, a predicate (relation), and a tail entity, such as (`Paris`, `is_in`, `France`). In this thesis, we focus on such triplet-based knowledge bases, but classical symbolic AI research has also studied knowledge bases that contain richer sentences with first or higher order logic (Lenat, 1995).

A triplet-based knowledge base can be represented as a multi-relational graph (**knowledge graph**) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of entity nodes in the KG and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ is the set of edges (triplets) that connect nodes in $\mathcal{V}$, with $\mathcal{R}$ being the set of relation types $\{r\}$. Each triplet (h, r, t) in a KG can represent a knowledge fact such as (`Paris`, `in`, `France`).

Examples of general knowledge bases include Freebase (Bollacker et al., 2008), ConceptNet (Speer et al., 2017), Wikidata (Vrandečić and Krötzsch, 2014), and domain-specific knowledge bases include UMLS (Bodenreider, 2004) and DrugBank (Wishart et al., 2018).

The benefits of knowledge bases lie in the following facets:

- Since knowledge bases explicitly store and organize facts, they provide interpretability, provenance, and ease of updating facts (Ahmed et al., 2019).
- Knowledge bases can provide structured information valuable for executing complex queries and performing logical reasoning (Hamilton et al., 2018; Ren et al., 2020).
- Knowledge bases offer domain expertise, as professionals in specialized fields such as biology, medicine, and finance often curate domain knowledge within them (Hewett et al., 2002).

Model

Graph Neural Networks (GNNs; Hamilton et al. 2017; Xu et al. 2018) are often used to encode graph structured data, including knowledge graphs, into contextualized vector representations. Given the graph G, GNNs produce the vector representation of each node by aggregating and transforming the representations of its neighbors from the previous layer. Specifically, using $\mathbf{h}_v^{(\ell)}$ to denote the representation of node v at the ℓ-th layer of a GNN, each GNN layer updates the node representations

by:

$$\mathbf{a}_v^{(\ell)} \leftarrow \text{AGGREGATE}^{(\ell)}(\{\mathbf{h}_u^{(\ell-1)} : u \in \mathcal{N}(v)\}), \tag{2.1}$$

$$\mathbf{h}_v^{(\ell)} \leftarrow \text{COMBINE}^{(\ell)}(\mathbf{h}_v^{(\ell-1)}, \mathbf{a}_v^{(\ell)}), \tag{2.2}$$

where $\mathcal{N}(v)$ is the set of neighbor nodes of v, and $\text{AGGREGATE}^{(\ell)}(\cdot)$ and $\text{COMBINE}^{(\ell)}(\cdot)$ are functions modeled using neural networks, such as sum-pooling followed by a Multilayer perceptron (MLP).

As evident, a knowledge graph presents a distinct representation compared to text. In Chapter 4, we will extend language models to integrate both knowledge graph and text representations. Specifically, we will use GNNs to extract contextualized vector representations from the knowledge graph and fuse them with the language model's contextualized representations. We will discuss additional related works on knowledge graphs in §4.2.

2.1.3 Visual knowledge

Source data

Our world inherently encompasses multimodal information, particularly visual and language data. Indeed, visuals convey the richest information in human communication and perception, complementing language (Burmark, 2002). For example, visuals can elucidate geometry, shape, and color more effectively than verbal descriptions, offering insights that textual expression cannot capture. In particular, images serve as a pivotal source of visual information, playing significant roles in multimodal applications. These applications include generating images from textual descriptions and answering questions about images, which are becoming increasingly more prevalent (Rombach et al., 2022; OpenAI, 2023).

Images are commonly represented as tensors of pixel values $I \in \mathbb{R}^{H \times W \times C}$, where H, W, and C denote the height, width, and number of channels (e.g., RGB) of the image and the pixel values indicate the color intensity.

Model

In deep learning models, images are commonly encoded by Convolutional Neural Networks (CNNs; Krizhevsky et al. 2012) into contextualized tensor representations. CNNs are designed to capture the spatial hierarchy of images. In the each convolutional layer, filters that perform convolution operations on the input image to extract features that are useful for capturing patterns such as edges, textures, and shapes.

More recently, image tokenization techniques have gained prominence, exemplified by the Vector

Quantized Variational Autoencoder (VQ-VAE; Van Den Oord et al. 2017) and the Vector Quantized Generative Adversarial Network (VQ-GAN; Esser et al. 2021). These models are designed to extract features from an image in the form of a token sequence. Specifically, the input image is first transformed into a tensor representation using encoders like CNN. This tensor is then reshaped into a sequence of vectors, each of which is quantized by finding the closest vector in a predefined dictionary or codebook, resulting in a sequence of tokens. These token sequences can then be further processed by powerful Transformer models.

Images, in their raw form, offer a distinct representation compared to text. In Chapter 5, we will extend language models to integrate both image and text representations. Specifically, we will employ the aforementioned image tokenization techniques, such as VQ-GAN, to convert images into sequences of tokens. This approach enables the unified handling of interleaved sequences of text and images as sequences of tokens. We will discuss additional related works on vision-language models in §5.2.

2.1.4 Knowledge retrieval

Given a query, the trained model itself may not have the knowledge to answer it. Therefore, the model should be able to retrieve relevant knowledge from various sources, much like how humans browse the web to find and confirm answers (Chen et al., 2017). For example, the model should be able to retrieve relevant documents from the web texts, relevant knowledge subgraphs from large-scale knowledge graphs, and relevant images from web images to respond to the query and perform the task.

In text retrieval, typical techniques involve computing the relevance score between the query and the candidate document, either using sparse lexical features like bag-of-words (sparse retrieval, such as TF-IDF and BM25; Robertson et al. 2009) or dense features produced by neural text encoders (dense retrieval; Karpukhin et al. 2020). Knowledge graph retrieval typically involves performing entity linking of the query and then considering the neighboring entities in the knowledge graph. Image retrieval from text query can be accomplished by performing dense retrieval based on contrastively trained text encoders and image encoders (Radford et al., 2021b). We will discuss the specific methods we use to retrieve text, knowledge graphs and images in Chapter 3, 4, and 5, respectively.

2.2 Historical Context

How to represent knowledge and use knowledge to perform tasks has been a long-standing area of research in AI. We review the historical context surrounding research in knowledge and AI and discuss the positioning of this thesis within it.

2.2.1 Traditional knowledge base systems

In the history of AI, knowledge base systems emerged since the 1970s as foundational constructs for representing and reasoning about knowledge (Hayes-Roth et al., 1983). These systems were created to capture, organize, and use knowledge in a structured manner, facilitating automated reasoning for applications like NLP, question answering, expert systems, and database management.

Initially, knowledge base systems focused on abstract models for knowledge representation and reasoning, typically dealing with small-scale knowledge bases (KBs). These systems typically included the following facets (Levesque, 1986; Newell and Simon, 2007):

- Language design: Creating a formal language, such as first-order logic, to represent knowledge.
- Ontology development: Establishing an object model for structured information organization, encompassing classes, subclasses, properties, relationships, and instances. For example, an instance of the class 'Book' contains properties like 'title' and relationships like 'authored by', linking to an instance of the class 'Author'.
- Knowledge base construction: Curating logical statements to represent facts. For instance, expressing facts like Book(HarryPotter) $\wedge$ Author(JKRowling) $\wedge$ AuthoredBy(HarryPotter, JKRowling).
- Inference: These systems also developed inference mechanisms to translate input text (queries) into logical forms and execute queries against the KB to obtain answers (Woods, 1972; Winograd, 1972).

As the field progresses, efforts were made to create general-purpose large-scale KBs, especially in the domain of commonsense background knowledge. Two notable examples are Cyc and ConceptNet.

- Cyc (Lenat, 1995) was developed with the goal of capturing comprehensive commonsense knowledge about the world in a machine-readable format. It designed an expressive representation language, CycL, that extends beyond first-order to higher-order logic; developed an ontology that spans all human concepts, providing detail down to an appropriate level; manually curated logical statements on top of the ontology to capture all human knowledge about those concepts in a detailed manner; and developed an inference engine that was exponentially faster than those employed in conventional expert systems at the time. This aimed to enable the system to infer conclusions of comparable types and depth to those achievable by humans. However, Cyc faced the scalability challenge due to the system's complexity and the high cost of knowledge curation.
- More recent knowledge bases like ConceptNet (Speer et al., 2017) aimed to capture commonsense knowledge about the world by employing simpler knowledge representation and semi-automatic curation techniques. ConceptNet is expressed as a directed graph where nodes

represent concepts, and edges represent assertions of commonsense pertaining to these concepts (concept-relation-concept triplets). Concepts denote sets of closely related natural language phrases, encompassing noun phrases, verb phrases, adjective phrases, or clauses. The knowledge base is populated by extracting natural language assertions or concept-relation-concept triplets from text corpora, utilizing "fill-in-the-blanks" templates alongside human verification.

The strength of these knowledge base systems lies in their ability to solve complex inference problems precisely by harnessing the power of logic, provided that the query and requisite knowledge can be articulated within the language and inference framework of the knowledge base. However, traditional knowledge base systems also have notable limitations. They heavily depend on the predetermined logical representations, but not all types of knowledge can be precisely expressed through logical forms and they also struggle to handle the nuances, context dependencies, and degree of certainty that can be expressed in natural language. For instance, consider the statement, "John usually prefers chocolate ice cream, but he might opt for vanilla on hot days." Moreover, traditional knowledge bases often rely on manual curation of knowledge statements, making it challenging to scale beyond specific domains.

2.2.2 Statistical machine learning and deep learning

Since the 1990s, machine learning has revolutionized natural language processing (NLP) by introducing a paradigm shift: collecting examples of desired input-output behaviors, such as question-answer pairs, to train statistical models. Here, these input-output examples can be viewed as providing the knowledge for performing tasks, and the trained model can be viewed as the inference engine. As this approach is simple and scalable due to the increasing availability of data and computational resources, it enabled machine learning to excel in various NLP tasks (Berger et al., 1996; Joachims, 1998; Zelle and Mooney, 1996; Miller et al., 1996; Toutanova et al., 2003; Klein and Manning, 2003; Collins, 2003; Wiebe et al., 2005; Ott et al., 2011; Zettlemoyer and Collins, 2012; Liang et al., 2013; Berant et al., 2013).

The popularity of machine learning surged further in the 2010s with the rise of deep learning. Neural network models, characterized by their expressive representations produced by deep model architectures, have been developed for many NLP tasks, including natural language inference, question answering, text classification, sentiment analysis, and text generation (Bowman et al., 2015; Chen et al., 2016; Parikh et al., 2016; Hermann et al., 2015; Seo et al., 2016; Kim, 2014; Socher et al., 2013; Sutskever et al., 2014; Bahdanau et al., 2014). Reading comprehension tasks also gained traction, where the systems are tasked to read documents to generate accurate answers (Rajpurkar et al., 2016; Chen et al., 2017; Yang et al., 2018), and here text is considered as the source of knowledge.

More recently, around 2017, a new paradigm known as pretraining gained prominence. Unlike traditional machine learning approaches that train specific models on task-specific datasets, the pretraining paradigm involves training large language models on vast corpora of textual data from

diverse sources (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020; Radford et al., 2019; Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023). A key advantage of pretraining is its ability to generalize across various tasks at the test time, rather than being limited to specific tasks (Bommasani et al., 2021; Liang et al., 2022; Qin et al., 2023).

However, despite these advancements, several limitations persist. Existing models often struggle with knowledge-intensive tasks that demand access to the latest, complex, or domain-specific expert knowledge. Moreover, these models primarily focus on textual data and lack the capability to incorporate modalities beyond text, such as visual information, limiting their scope of applications. Consequently, there is a pressing need to develop systems that effectively integrate diverse forms and modalities of knowledge to perform a wider range of tasks.

2.2.3 This thesis

Given the historical context outlined above, this thesis aims to develop end-to-end models capable of using diverse forms of knowledge to facilitate question answering. In doing so, we aim to alleviate the limitations inherent in traditional knowledge base systems and conventional machine learning approaches.

Specifically, regarding knowledge representation, previous approaches face challenges as not all knowledge can be expressed solely through logical statements (§2.2.1) or textual formats (§2.2.2). While text offers better flexibility, it remains insufficient due to the existence of other forms of information, such as visual knowledge. In this thesis, we will address this limitation by covering all forms of knowledge representation, including knowledge bases (KBs), text, and images, in their original formats. By using these different modalities, we aim to leverage their complementary strengths; for instance, text provides broad coverage, KBs offer structured and curated information, and images introduce an additional modality.

Regarding inference, or answering queries, a similar challenge arises in previous approaches, as not all queries can be expressed solely through logical forms or text; for instance, users may wish to pose questions related to images. Therefore, this thesis will focus on developing models that can use a combination of text, KBs, and images as both knowledge sources and queries in an end-to-end, soft manner. We will achieve this by developing unified architectures that integrate techniques such as Transformers, Graph Neural Networks (GNNs), and image tokenizers.

Part I

Methodologies

In this part, we present the core methodologies for effectively fusing textual knowledge (Chapter 3), structured knowledge (Chapter 4), and visual knowledge (Chapter 5) into language models.

Chapter 3

Fusing Textual Knowledge

Textual data provides broad and contextually-rich knowledge. In this chapter, we introduce methods for language models to efficiently learn knowledge from textual data.

3.1 Introduction

Pretrained language models (LMs), like BERT and GPTs (Devlin et al., 2019; Brown et al., 2020), have shown remarkable performance on many natural language processing (NLP) tasks, such as text classification and question answering, becoming the foundation of modern NLP systems (Bommasani et al., 2021). By performing self-supervised learning, such as masked language modeling (Devlin et al., 2019), LMs learn to encode various knowledge from text corpora and produce informative representations for downstream tasks (Petroni et al., 2019; Bosselut et al., 2019; Raffel et al., 2020).

However, existing LM pretraining methods typically consider text from a single document in each input context (Liu et al., 2019; Joshi et al., 2020) and do not model links between documents. This can pose limitations because documents often have rich dependencies (e.g. hyperlinks, references), and knowledge can span *across* documents. As an example, in Figure 3.1, the Wikipedia article "Tidal Basin, Washington D.C." (left) describes that the basin hosts "National Cherry Blossom Festival", and the hyperlinked article (right) reveals the background that the festival celebrates "Japanese cherry trees". Taken together, the hyperlink offers new, multi-hop knowledge "Tidal Basin has Japanese cherry trees", which is not available in the single article "Tidal Basin" alone. Acquiring such multi-hop knowledge in pretraining could be useful for various applications including question answering. In fact, document links like hyperlinks and references are ubiquitous (e.g. web, books, scientific literature), and guide how we humans acquire knowledge and even make discoveries (Margolis et al., 1999).

In this work, we propose *LinkBERT*, an effective language model pretraining method that incorporates document link knowledge. Given a text corpus, we obtain links between documents such

as hyperlinks, and create LM inputs by placing linked documents in the same context, besides the existing option of placing a single document or random documents as in BERT. Specifically, as in Figure 3.2, after sampling an anchor text segment, we place either (1) the contiguous segment from the same document, (2) a random document, or (3) a document linked from anchor segment, as the next segment in the input. We then train the LM with two joint objectives: We use masked language modeling (MLM) to encourage learning multi-hop knowledge of concepts brought into the same context by document links (e.g. "Tidal Basin" and "Japanese cherry" in Figure 3.1). Simultaneously, we propose a Document Relation Prediction (DRP) objective, which classifies the relation of the second segment to the first segment (*contiguous*, *random*, or *linked*). DRP encourages learning the relevance and bridging concepts (e.g. "National Cherry Blossom Festival") between documents, beyond the ability learned in the vanilla next sentence prediction objective in BERT.

Viewing the pretraining corpus as a graph of documents, LinkBERT is also motivated as self-supervised learning on the graph, where DRP and MLM correspond to link prediction and node feature prediction in graph machine learning (Yang et al., 2015; Hu et al., 2020). Our modeling approach thus provides a natural fusion of language-based and graph-based self-supervised learning.

We train LinkBERT in two domains: the general domain, using Wikipedia articles with hyperlinks (§3.3), and the biomedical domain, using PubMed articles with citation links (§3.5). We then evaluate the pretrained models on a wide range of downstream tasks such as question answering, in both domains. LinkBERT consistently improves on baseline LMs across domains and tasks. For the general domain, LinkBERT outperforms BERT on MRQA benchmark (+4% absolute in F1-score) as well as GLUE benchmark. For the biomedical domain, LinkBERT exceeds PubmedBERT (Gu et al., 2021) and sets new states of the art on BLURB biomedical NLP benchmark (+3% absolute in BLURB score) and MedQA-USMLE reasoning task (+7% absolute in accuracy). Overall, LinkBERT attains notably large gains for multi-hop reasoning, multi-document understanding, and few-shot question answering, suggesting that LinkBERT internalizes significantly more knowledge than existing LMs by pretraining with document link information.

Our pretrained models, *LinkBERT* and *BioLinkBERT*, are available at `https://github.com/michiyasunaga/LinkBERT`.

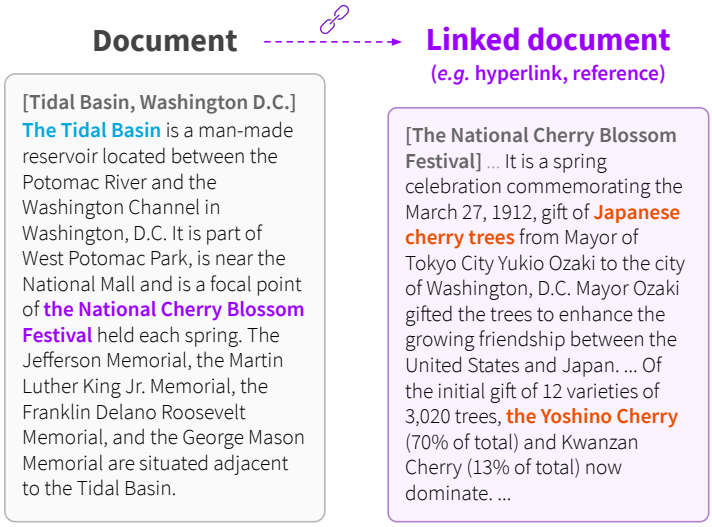


Figure 3.1: Document links (e.g. hyperlinks) can provide salient multi-hop knowledge. For instance, the Wikipedia article "Tidal Basin" (left) describes that the basin hosts "National Cherry Blossom Festival". The hyperlinked article (right) reveals that the festival celebrates "Japanese cherry trees". Taken together, the link suggests new knowledge not available in a single document (e.g. "Tidal Basin has Japanese cherry trees"), which can be useful for various applications, including answering a question "What trees can you see at Tidal Basin?". We aim to leverage document links to incorporate more knowledge into language model pretraining.

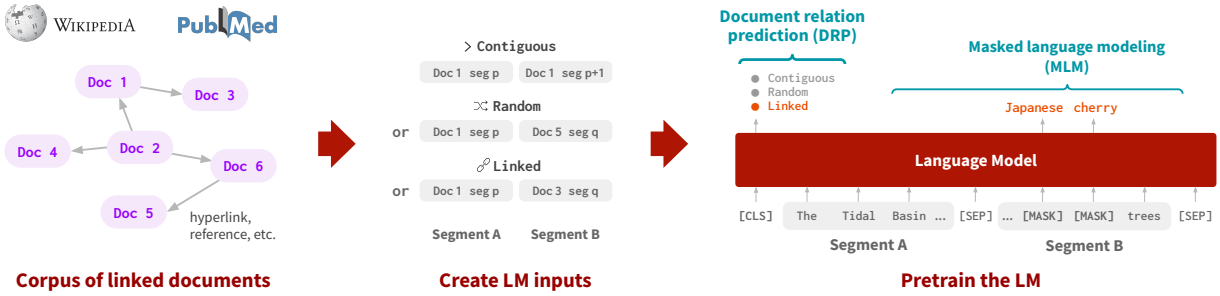


Figure 3.2: **Overview of our approach, LinkBERT**. Given a pretraining corpus, we view it as a graph of documents, with links such as hyperlinks (§3.3.1). To incorporate the document link knowledge into LM pretraining, we create LM inputs by placing a pair of linked documents in the same context (*linked*), besides the existing options of placing a single document (*contiguous*) or a pair of random documents (*random*) as in BERT. We then train the LM with two self-supervised objectives: masked language modeling (MLM), which predicts masked tokens in the input, and document relation prediction (DRP), which classifies the relation of the two text segments in the input (*contiguous*, *random*, or *linked*) (§3.3.2).

3.2 Preliminaries

A language model (LM) can be pretrained from a corpus of documents, $\mathcal{X} = \{X^{(i)}\}$. An LM is a composition of two functions, $f_{\text{head}}(f_{\text{enc}}(X))$, where the encoder f_{enc} takes in a sequence of tokens $X = (x_1, x_2, ..., x_n)$ and produces a contextualized vector representation for each token, $(\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n)$. The head f_{head} uses these representations to perform self-supervised tasks in the pretraining step and to perform downstream tasks in the fine-tuning step. We build on BERT (Devlin et al., 2019), which pretrains an LM with the following two self-supervised tasks.

Masked language modeling (MLM). Given a sequence of tokens X, a subset of tokens $Y \subseteq X$ is masked, and the task is to predict the original tokens from the modified input. Y accounts for 15% of the tokens in X; of those, 80% are replaced with `[MASK]`, 10% with a random token, and 10% are kept unchanged.

Next sentence prediction (NSP). The NSP task takes two text segments[1] (X_A, X_B) as input, and predicts whether X_B is the direct continuation of X_A. Specifically, BERT first samples X_A from the corpus, and then either (1) takes the next segment X_B from the same document, or (2) samples X_B from a random document in the corpus. The two segments are joined via special tokens to form an input instance, `[CLS]` X_A `[SEP]` X_B `[SEP]`, where the prediction target of `[CLS]` is whether X_B indeed follows X_A (*contiguous* or *random*).

In this work, we will further incorporate document link information into LM pretraining. Our approach (§3.3) will build on MLM and NSP.

3.3 Approach

We present LinkBERT, a self-supervised pretraining approach that aims to internalize more knowledge into LMs using document link information. Specifically, as shown in Figure 3.2, instead of viewing the pretraining corpus as a set of documents $\mathcal{X} = \{X^{(i)}\}$, we view it as a *graph* of documents, $\mathcal{G} = (\mathcal{X}, \mathcal{E})$, where $\mathcal{E} = \{(X^{(i)}, X^{(j)})\}$ denotes links between documents (§3.3.1). The links can be existing hyperlinks, or could be built by other methods that capture document relevance. We then consider pretraining tasks for learning from document links (§3.3.2): We create LM inputs by placing linked documents in the same context window, besides the existing options of a single document or random documents. We use the MLM task to learn concepts brought together in the context by document links, and we also introduce the Document Relation Prediction (DRP) task to learn relations between documents. Finally, we discuss strategies for obtaining informative pairs of linked documents to feed into LM pretraining (§3.3.3).

[1] A segment is typically a sentence or a paragraph.

3.3.1 Document graph

Given a pretraining corpus, we link related documents so that the links can bring together knowledge that is not available in single documents. We focus on hyperlinks, e.g., hyperlinks of Wikipedia articles (§3.4) and citation links of academic articles (§3.5). Hyperlinks have a number of advantages. They provide background knowledge about concepts that the document writers deemed useful—the links are likely to have high precision of relevance, and can also bring in relevant documents that may not be obvious via lexical similarity alone (e.g., in Figure 3.1, while the hyperlinked article mentions "Japanese" and "Yoshino" cherry trees, these words do not appear in the anchor article). Hyperlinks are also ubiquitous on the web and easily gathered at scale (Aghajanyan et al., 2021). To construct the document graph, we simply make a directed edge $(X^{(i)}, X^{(j)})$ if there is a hyperlink from document $X^{(i)}$ to document $X^{(j)}$.

For comparison, we also experiment with a document graph built by lexical similarity between documents. For each document $X^{(i)}$, we use the common TF-IDF cosine similarity metric (Chen et al., 2017; Yasunaga et al., 2017) to obtain top-k documents $X^{(j)}$'s and make edges $(X^{(i)}, X^{(j)})$. We use $k = 5$.

3.3.2 Pretraining tasks

Creating input instances. Several works (Gao et al., 2021; Levine et al., 2021) find that LMs can learn stronger dependencies between words that were shown together in the same context during training, than words that were not. To effectively learn knowledge that spans across documents, we create LM inputs by placing linked documents in the same context window, besides the existing option of a single document or random documents. Specifically, we first sample an anchor text segment from the corpus (Segment A; $X_A \subseteq X^{(i)}$). For the next segment (Segment B; X_B), we either (1) use the contiguous segment from the same document ($X_B \subseteq X^{(i)}$), (2) sample a segment from a random document ($X_B \subseteq X^{(j)}$ where $j \neq i$), or (3) sample a segment from one of the documents linked from Segment A ($X_B \subseteq X^{(j)}$ where $(X^{(i)}, X^{(j)}) \in \mathcal{E}$). We then join the two segments via special tokens to form an input instance: `[CLS]` X_A `[SEP]` X_B `[SEP]`.

Training objectives. To train the LM, we use two objectives. The first is the MLM objective to encourage the LM to learn multi-hop knowledge of concepts brought into the same context by document links. The second objective, which we propose, is Document Relation Prediction (DPR), which classifies the relation r of segment X_B to segment X_A ($r \in \{contiguous, random, linked\}$). By distinguishing *linked* from *contiguous* and *random*, DRP encourages the LM to learn the relevance and existence of bridging concepts between documents, besides the capability learned in the vanilla NSP objective. To predict r, we use the representation of `[CLS]` token, as in NSP. Taken together,

we optimize:

$$\mathcal{L} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{DRP}} \tag{3.1}$$

$$= -\sum_i \log\ p(x_i \mid \mathbf{h}_i) - \log\ p(r \mid \mathbf{h}_{\texttt{[CLS]}}) \tag{3.2}$$

where x_i is each token of the input instance, `[CLS]` X_A `[SEP]` X_B `[SEP]`, and $\mathbf{h}_i$ is its representation.

Graph machine learning perspective. Our two pretraining tasks, MLM and DRP, are also motivated as graph self-supervised learning on the document graph. In graph self-supervised learning, two types of tasks, node feature prediction and link prediction, are commonly used to learn the content and structure of a graph. In node feature prediction (Hu et al., 2020), some features of a node are masked, and the task is to predict them using neighbor nodes. This corresponds to our MLM task, where masked tokens in Segment A can be predicted using Segment B (a linked document on the graph), and vice versa. In link prediction (Bordes et al., 2013; Wang et al., 2021a), the task is to predict the existence or type of an edge between two nodes. This corresponds to our DRP task, where we predict if the given pair of text segments are linked (edge), contiguous (self-loop edge), or random (no edge). Our approach can be viewed as a natural fusion of language-based (e.g. BERT) and graph-based self-supervised learning.

3.3.3 Strategy to obtain linked documents

As described in §3.3.1, §3.3.2, our method *builds* links between documents, and for each anchor segment, *samples* a linked document to put together in the LM input. Here we discuss three key axes to consider to obtain useful linked documents in this process.

Relevance. Semantic relevance is a requisite when building links between documents. If links were randomly built without relevance, LinkBERT would be same as BERT, with simply two options of LM inputs (*contiguous* or *random*). Relevance can be achieved by using hyperlinks or lexical similarity metrics, and both methods yield substantially better performance than using random links (§3.4.5).

Salience. Besides relevance, another factor to consider (*salience*) is whether the linked document can offer new, useful knowledge that may not be obvious to the current LM. Hyperlinks are potentially more advantageous than lexical similarity links in this regard: LMs are shown to be good at recognizing lexical similarity (Zhang et al., 2020), and hyperlinks can bring in useful background knowledge that may not be obvious via lexical similarity alone (Asai et al., 2020). Indeed, we empirically find that using hyperlinks yields a more performant LM (§3.4.5).

Diversity. In the document graph, some documents may have a very high in-degree (e.g., many incoming hyperlinks, like the "United States" page of Wikipedia), and others a low in-degree. If we uniformly sample from the linked documents for each anchor segment, we may include documents of high in-degree too often in the overall training data, losing diversity. To adjust so that all documents appear with a similar frequency in training, we sample a linked document with probability inversely proportional to its in-degree, as done in graph data mining literature (Henzinger et al., 2000). We find that this technique yields a better LM performance (§3.4.5).

3.4 Experiments: General domain

We experiment with our proposed approach in the general domain first, where we pretrain LinkBERT on Wikipedia articles with hyperlinks (§3.4.1) and evaluate on a suite of downstream tasks (§3.4.2). We compare with BERT (Devlin et al., 2019) as our baseline. We experiment in the biomedical domain in §3.5.

3.4.1 Pretraining setup

Data. We use the same pretraining corpus used by BERT: Wikipedia and BookCorpus (Zhu et al., 2015). For Wikipedia, we use the WikiExtractor[2] to extract hyperlinks between Wiki articles. We then create training instances by sampling *contiguous*, *random*, or *linked* segments as described in §3.3, with the three options appearing uniformly (33%, 33%, 33%). For BookCorpus, we create training instance by sampling *contiguous* or *random* segments (50%, 50%) as in BERT. We then combine the training instances from Wikipedia and BookCorpus to train LinkBERT. In summary, our pretraining data is the same as BERT, except that we have hyperlinks between Wikipedia articles.

Implementation. We pretrain LinkBERT of three sizes, -tiny, -base and -large, following the configurations of $\text{BERT}_{\texttt{tiny}}$ (4.4M parameters), $\text{BERT}_{\texttt{base}}$ (110M params), and $\text{BERT}_{\texttt{large}}$ (340M params) (Devlin et al., 2019; Turc et al., 2019). We use -tiny mainly for ablation studies.

For -tiny, we pretrain from scratch with random weight initialization. We use the AdamW (Loshchilov and Hutter, 2019) optimizer with $(\beta_1, \beta_2) = (0.9, 0.98)$, warm up the learning rate for the first 5,000 steps and then linearly decay it. We train for 10,000 steps with a peak learning rate 5e-3, weight decay 0.01, and batch size of 2,048 sequences with 512 tokens. Training took 1 day on two GeForce RTX 2080 Ti GPUs with fp16.

For -base, we initialize LinkBERT with the $\text{BERT}_{\texttt{base}}$ checkpoint released by Devlin et al. (2019) and continue pretraining. We use a peak learning rate 3e-4 and train for 40,000 steps. Other training hyperparameters are the same as -tiny. Training took 4 days on four A100 GPUs with fp16.

[2] https://github.com/attardi/wikiextractor

For -large, we follow the same procedure as -base, except that we use a peak learning rate of 2e-4. Training took 7 days on eight A100 GPUs with fp16.

Baselines. We compare LinkBERT with BERT. Specifically, for the -tiny scale, we compare with $\text{BERT}_{\texttt{tiny}}$, which we pretrain from scratch with the same hyperparameters as $\text{LinkBERT}_{\texttt{tiny}}$. The only difference is that LinkBERT uses document links to create LM inputs, while BERT does not.

For -base scale, we compare with $\text{BERT}_{\texttt{base}}$, for which we take the $\text{BERT}_{\texttt{base}}$ release by Devlin et al. (2019) and continue pretraining it with the vanilla BERT objectives on the same corpus for the same number of steps as $\text{LinkBERT}_{\texttt{base}}$.

For -large, we follow the same procedure as -base.

3.4.2 Evaluation tasks

We fine-tune and evaluate LinkBERT on a suite of downstream tasks.

Extractive question answering (QA). Given a document (or set of documents) and a question as input, the task is to identify an answer span from the document. We evaluate on six popular datasets from the MRQA shared task (Fisch et al., 2019): *HotpotQA* (Yang et al., 2018), *TriviaQA* (Joshi et al., 2017), *NaturalQ* (Kwiatkowski et al., 2019), *SearchQA* (Dunn et al., 2017), *NewsQA* (Trischler et al., 2017), and *SQuAD* (Rajpurkar et al., 2016). As the MRQA shared task does not have a public test set, we split the dev set in half to make new dev and test sets. We follow the fine-tuning method BERT (Devlin et al., 2019) uses for extractive QA. More details are provided in §3.8.

GLUE. The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) is a popular suite of sentence-level classification tasks. Following BERT, we evaluate on *CoLA* (Warstadt et al., 2019), *SST-2* (Socher et al., 2013), *MRPC* (Dolan and Brockett, 2005), *QQP*, *STS-B* (Cer et al., 2017), *MNLI* (Williams et al., 2017), *QNLI* (Rajpurkar et al., 2016), and *RTE* (Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007), and report the average score. More fine-tuning details are provided in §3.8.

3.4.3 Results

Table 3.1 shows the performance (F1 score) on MRQA datasets. LinkBERT substantially outperforms BERT on all datasets. On average, the gain is +4.1% absolute for the $\text{BERT}_{\texttt{tiny}}$ scale, +2.6% for the $\text{BERT}_{\texttt{base}}$ scale, and +2.5% for the $\text{BERT}_{\texttt{large}}$ scale. Table 3.2 shows the results on GLUE, where LinkBERT performs moderately better than BERT. These results suggest that LinkBERT is especially effective at learning knowledge useful for QA tasks (e.g. world knowledge), while keeping performance on sentence-level language understanding.

	HotpotQA	TriviaQA	SearchQA	NaturalQ	NewsQA	SQuAD	Avg.
$\text{BERT}_{\texttt{tiny}}$	49.8	43.4	50.2	58.9	41.3	56.6	50.0
$\text{LinkBERT}_{\texttt{tiny}}$	**54.6**	**50.0**	**58.6**	**60.3**	**42.8**	**58.0**	**54.1**
$\text{BERT}_{\texttt{base}}$	76.0	70.3	74.2	76.5	65.7	88.7	75.2
$\text{LinkBERT}_{\texttt{base}}$	**78.2**	**73.9**	**76.8**	**78.3**	**69.3**	**90.1**	**77.8**
$\text{BERT}_{\texttt{large}}$	78.1	73.7	78.3	79.0	70.9	91.1	78.5
$\text{LinkBERT}_{\texttt{large}}$	**80.8**	**78.2**	**80.5**	**81.0**	**72.6**	**92.7**	**81.0**

Table 3.1: Performance (F1) on MRQA question answering datasets. LinkBERT consistently outperforms BERT on all datasets across the -tiny, -base, and -large scales. The gain is especially large on datasets that require reasoning with multiple documents in the context, such as HotpotQA, TriviaQA, SearchQA.

	GLUE score
$\text{BERT}_{\texttt{tiny}}$	64.3
$\text{LinkBERT}_{\texttt{tiny}}$	**64.6**
$\text{BERT}_{\texttt{base}}$	79.2
$\text{LinkBERT}_{\texttt{base}}$	**79.6**
$\text{BERT}_{\texttt{large}}$	80.7
$\text{LinkBERT}_{\texttt{large}}$	**81.1**

Table 3.2: Performance on the GLUE benchmark. LinkBERT attains comparable or moderately improved performance.

	SQuAD	SQuAD distract
$\text{BERT}_{\texttt{base}}$	88.7	85.9
$\text{LinkBERT}_{\texttt{base}}$	**90.1**	**89.6**

Table 3.3: Performance (F1) on SQuAD when distracting documents are added to the context. While BERT incurs a large drop in F1, LinkBERT does not, suggesting its robustness in understanding document relations.

	HotpotQA	TriviaQA	NaturalQ	SQuAD
$\text{BERT}_{\texttt{base}}$	64.8	59.2	64.8	79.6
$\text{LinkBERT}_{\texttt{base}}$	**70.5**	**66.0**	**70.2**	**82.8**

Table 3.4: Few-shot QA performance (F1) when 10% of fine-tuning data is used. LinkBERT attains large gains, suggesting that it internalizes more knowledge than BERT in pretraining.

	HotpotQA	TriviaQA	NaturalQ	SQuAD
LinkBERT$_{\texttt{tiny}}$	**54.6**	**50.0**	**60.3**	**58.0**
No diversity	53.5	48.0	60.0	57.8
Change hyperlink to TF-IDF	50.0	48.2	59.6	57.6
Change hyperlink to random	49.8	43.4	58.9	56.6

Table 3.5: Ablation study on what linked documents to feed into LM pretraining (§3.3.3).

	HotpotQA	TriviaQA	NaturalQ	SQuAD	SQuAD distract
LinkBERT$_{\texttt{base}}$	**78.2**	**73.9**	**78.3**	**90.1**	**89.6**
No DRP	76.5	72.5	77.0	89.3	87.0

Table 3.6: Ablation study on the document relation prediction (DRP) objective in LM pretraining (§3.3.2).

3.4.4 Analysis

We further study when LinkBERT is especially useful in downstream tasks.

Improved multi-hop reasoning. In Table 3.1, we find that LinkBERT obtains notably large gains on QA datasets that require reasoning with multiple documents, such as HotpotQA (+5% over $\text{BERT}_{\texttt{tiny}}$), TriviaQA (+6%) and SearchQA (+8%), as opposed to SQuAD (+1.4%) which just has a single document per question. To further gain qualitative insights, we studied in what QA examples LinkBERT succeeds but BERT fails. Figure 3.3 shows a representative example from HotpotQA. Answering the question needs 2-hop reasoning: identify "Roden Brothers were taken over by Birks Group" from the first document, and then "Birks Group is headquartered in Montreal" from the second document. While BERT tends to simply predict an entity near the question entity ("Toronto" in the first document, which is just 1-hop), LinkBERT correctly predicts the answer in the second document ("Montreal"). Our intuition is that because LinkBERT is pretrained with pairs of linked documents rather than purely single documents, it better learns how to flow information (e.g., do attention) across tokens when multiple related documents are given in the context. In summary, these results suggest that pretraining with linked documents helps for multi-hop reasoning on downstream tasks.

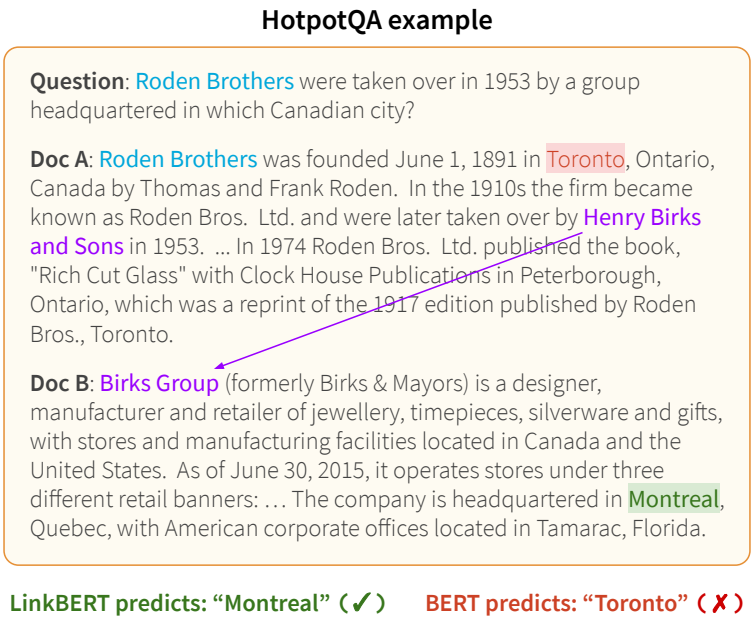


Figure 3.3: Case study of multi-hop reasoning on HotpotQA. Answering the question needs to identify "Roden Brothers were taken over by Birks Group" from the first document, and then "Birks Group is headquartered in Montreal" from the second document. While BERT tends to simply predict an entity near the question entity ("Toronto" in the first document), LinkBERT correctly predicts the answer in the second document ("Montreal").

Improved understanding of document relations. While the MRQA datasets typically use ground-truth documents as context for answering questions, in open-domain QA, QA systems need to use documents obtained by a retriever, which may include noisy documents besides gold ones (Chen et al., 2017; Dunn et al., 2017). In such cases, QA systems need to understand the document relations to perform well (Yang et al., 2018). To simulate this setting, we modify the SQuAD dataset by prepending or appending 1–2 distracting documents to the original document given to each question. Table 3.3 shows the result. While BERT incurs a large performance drop (-2.8%), LinkBERT is robust to distracting documents (-0.5%). This result suggests that pretraining with document links improves the ability to understand document relations and relevance. In particular, our intuition is that the DRP objective helps the LM to better recognize document relations like (anchor document, linked document) in pretraining, which helps to recognize relations like (question, right document) in downstream QA tasks. We indeed find that ablating the DRP objective from LinkBERT hurts performance (§3.4.5). The strength of understanding document relations also suggests the promise of applying LinkBERT to various retrieval-augmented methods and tasks (e.g. Lewis et al. 2020c), either as the main LM or the dense retriever component.

Improved few-shot QA performance. We also find that LinkBERT is notably good at few-shot learning. Concretely, for each MRQA dataset, we fine-tune with only 10% of the available training data, and report the performance in Table 3.4. In this few-shot regime, LinkBERT attains more significant gains over BERT, compared to the full-resource regime in Table 3.1 (on NaturalQ, 5.4% vs 1.8% absolute in F1, or 15% vs 7% in relative error reduction). This result suggests that LinkBERT internalizes more knowledge than BERT during pretraining, which supports our core idea that document links can bring in new, useful knowledge for LMs.

3.4.5 Ablation studies

We conduct ablation studies on the key design choices of LinkBERT.

What linked documents to feed into LMs? We study the strategies discussed in §3.3.3 for obtaining linked documents: relevance, salience, and diversity. Table 3.5 shows the ablation result on MRQA datasets. First, if we ignore relevance and use random document links instead of hyperlinks, we get the same performance as BERT (-4.1% on average; "random" in Table 3.5). Second, using lexical similarity links instead of hyperlinks leads to 1.8% performance drop ("TF-IDF"). Our intuition is that hyperlinks can provide more salient knowledge that may not be obvious from lexical similarity alone. Nevertheless, using lexical similarity links is substantially better than BERT (+2.3%), confirming the efficacy of placing relevant documents together in the input for LM pretraining. Finally, removing the diversity adjustment in document sampling leads to 1% performance drop ("No diversity"). In summary, our insight is that to create informative inputs for LM pretraining,

the linked documents must be semantically relevant and ideally be salient and diverse.

Effect of the DRP objective. Table 3.6 shows the ablation result on the DRP objective (§3.3.2). Removing DRP in pretraining hurts downstream QA performance. The drop is large on tasks with multiple documents (HotpotQA, TriviaQA, and SQuAD with distracting documents). This suggests that DRP facilitates LMs to learn document relations.

3.5 Experiments: Biomedical domain

Pretraining LMs on biomedical text is shown to boost performance on biomedical NLP tasks (Beltagy et al., 2019; Lee et al., 2020; Lewis et al., 2020b; Gu et al., 2021). Biomedical LMs are typically trained on PubMed, which contains abstracts and citations of biomedical papers. While prior works only use their raw text for pretraining, academic papers have rich dependencies with each other via citations (references). We hypothesize that incorporating citation links can help LMs learn dependencies between papers and knowledge that spans across them.

With this motivation, we pretrain LinkBERT on PubMed with citation links (§3.5.1), which we term *BioLinkBERT*, and evaluate on biomedical downstream tasks (§3.5.2). As our baseline, we follow and compare with the state-of-the-art biomedical LM, PubmedBERT (Gu et al., 2021), which has the same architecture as BERT and is trained on PubMed.

3.5.1 Pretraining setup

Data. We use the same pretraining corpus used by PubmedBERT: PubMed abstracts (21GB).[3] We use the Pubmed Parser[4] to extract citation links between articles. We then create training instances by sampling *contiguous*, *random*, or *linked* segments as described in §3.3, with the three options appearing uniformly (33%, 33%, 33%). In summary, our pretraining data is the same as PubmedBERT, except that we have citation links between PubMed articles.

Implementation. We pretrain BioLinkBERT of -base size (110M params) from scratch, following the same hyperparamters as the PubmedBERT$_{\texttt{base}}$ (Gu et al., 2021). Specifically, we use a peak learning rate 6e-4, batch size 8,192, and train for 62,500 steps. We warm up the learning rate in the first 10% of steps and then linearly decay it. Training took 7 days on eight A100 GPUs with fp16.

Additionally, while the original PubmedBERT release did not include the -large size, we pretrain BioLinkBERT of the -large size (340M params) from scratch, following the same procedure as -base, except that we use a peak learning rate of 4e-4 and warm up steps of 20%. Training took 21 days on eight A100 GPUs with fp16.

[3] https://pubmed.ncbi.nlm.nih.gov. We use papers published before Feb. 2020 as in PubmedBERT.
[4] https://github.com/titipata/pubmed_parser

Baselines. We compare BioLinkBERT with PubmedBERT released by Gu et al. (2021).

3.5.2 Evaluation tasks

For downstream tasks, we evaluate on the BLURB benchmark (Gu et al., 2021), a diverse set of biomedical NLP datasets, and MedQA-USMLE (Jin et al., 2021b), a challenging biomedical QA dataset.

BLURB consists of five named entity recognition tasks, a PICO (population, intervention, comparison, and outcome) extraction task, three relation extraction tasks, a sentence similarity task, a document classification task, and two question answering tasks, as summarized in Table 3.7. We follow the same fine-tuning method and evaluation metric used by PubmedBERT (Gu et al., 2021).

MedQA-USMLE is a 4-way multi-choice QA task that tests biomedical and clinical knowledge. The questions are from practice tests for the US Medical License Exams (USMLE). The questions typically require multi-hop reasoning, e.g., given patient symptoms, infer the likely cause, and then answer the appropriate diagnosis procedure (Figure 3.4). We follow the fine-tuning method in Jin et al. (2021b). More details are provided in Appendix 3.8.

MMLU-professional medicine is a multi-choice QA task that tests biomedical knowledge and reasoning, and is part of the popular MMLU benchmark (Hendrycks et al., 2021) that is used to evaluate massive language models. We take the BioLinkBERT fine-tuned on the above MedQA-USMLE task, and evaluate on this task without further adaptation.

	PubMed-BERT$_{\texttt{base}}$	BioLink-BERT$_{\texttt{base}}$	BioLink-BERT$_{\texttt{large}}$
Named entity recognition			
BC5-chem (Li et al., 2016)	93.33	**93.75**	**94.04**
BC5-disease (Li et al., 2016)	85.62	**86.10**	**86.39**
NCBI-disease (Doğan et al., 2014)	87.82	**88.18**	**88.76**
BC2GM (Smith et al., 2008)	84.52	**84.90**	**85.18**
JNLPBA (Kim et al., 2004)	**80.06**	79.03	**80.06**
PICO extraction			
EBM PICO (Nye et al., 2018)	73.38	**73.97**	**74.19**
Relation extraction			
ChemProt (Krallinger et al., 2017)	77.24	**77.57**	**79.98**
DDI (Herrero-Zazo et al., 2013)	82.36	**82.72**	**83.35**
GAD (Bravo et al., 2015)	82.34	**84.39**	**84.90**
Sentence similarity			
BIOSSES (Soğancıoğlu et al., 2017)	92.30	**93.25**	**93.63**
Document classification			
HoC (Baker et al., 2016)	82.32	**84.35**	**84.87**
Question answering			
PubMedQA (Jin et al., 2019)	55.84	**70.20**	**72.18**
BioASQ (Nentidis et al., 2019)	87.56	**91.43**	**94.82**
BLURB score	81.10	**83.39**	**84.30**

Table 3.7: Performance on BLURB benchmark. BioLinkBERT attains improvement on all tasks, establishing new state of the art on BLURB. Gains are notably large on document-level tasks such as PubMedQA and BioASQ.

Methods	**Acc.** (%)
BioBERT$_{\texttt{large}}$ (Lee et al., 2020)	36.7
QAGNN (Yasunaga et al., 2021)	38.0
GreaseLM (Zhang et al., 2022b)	38.5
PubmedBERT$_{\texttt{base}}$ (Gu et al., 2021)	38.1
BioLinkBERT$_{\texttt{base}}$ (**Ours**)	**40.0**
BioLinkBERT$_{\texttt{large}}$ (**Ours**)	**44.6**

Table 3.8: Performance on MedQA-USMLE. BioLinkBERT outperforms all previous biomedical LMs.

Methods	**Acc.** (%)
GPT-3 (175B params) (Brown et al., 2020)	38.7
UnifiedQA (11B params) (Khashabi et al., 2020)	43.2
BioLinkBERT$_{\texttt{large}}$ (**Ours**)	**50.7**

Table 3.9: Performance on MMLU-professional medicine. BioLinkBERT significantly outperforms the largest general-domain LM or QA model, despite having just 340M parameters.

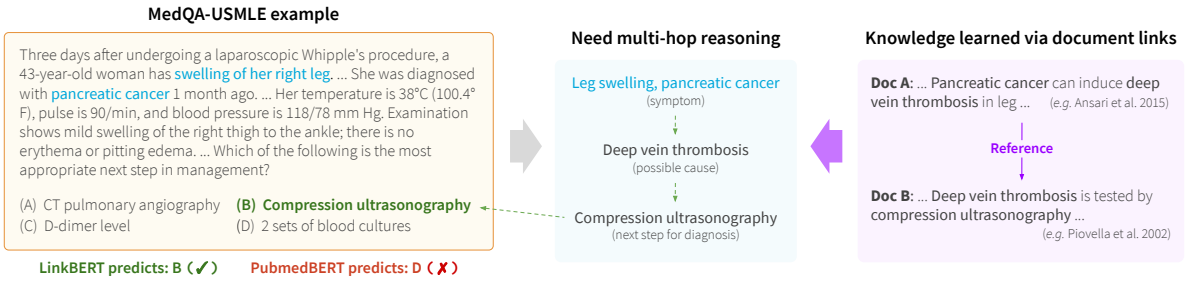


Figure 3.4: Case study of multi-hop reasoning on MedQA-USMLE. Answering the question (left) needs 2-hop reasoning (center): from the patient symptoms described in the question (*leg swelling*, *pancreatic cancer*), infer the cause (*deep vein thrombosis*), and then infer the appropriate diagnosis procedure (*compression ultrasonography*). While the existing PubmedBERT tends to simply predict a choice that contains a word appearing in the question ("blood" for choice D), BioLinkBERT correctly predicts the answer (B). Our intuition is that citation links bring relevant documents together in the same context in pretraining (right), which readily provides the multi-hop knowledge needed for the reasoning (center).

3.5.3 Results

BLURB. Table 3.7 shows the results on BLURB. BioLinkBERT$_{\texttt{base}}$ outperforms PubmedBERT$_{\texttt{base}}$ on all task categories, attaining a performance boost of +2% absolute on average. Moreover, BioLinkBERT$_{\texttt{large}}$ provides a further boost of +1%. In total, BioLinkBERT outperforms the previous best by +3% absolute, establishing a new state of the art on the BLURB leaderboard. We see a trend that gains are notably large on document-level tasks such as question answering (+7% on BioASQ and PubMedQA). This result is consistent with the general domain (§3.4.3) and confirms that LinkBERT helps to learn document dependencies better.

MedQA-USMLE. Table 3.8 shows the results. BioLinkBERT$_{\texttt{base}}$ obtains a 2% accuracy boost over PubmedBERT$_{\texttt{base}}$, and BioLinkBERT$_{\texttt{large}}$ provides an additional +5% boost. In total, BioLinkBERT outperforms the previous best by +7% absolute, setting a new state of the art. To further gain qualitative insights, we studied in what QA examples BioLinkBERT succeeds but the baseline PubmedBERT fails. Figure 3.4 shows a representative example. Answering the question (left) needs 2-hop reasoning (center): from the patient symptoms described in the question (*leg swelling*, *pancreatic cancer*), infer the cause (*deep vein thrombosis*), and then infer the appropriate diagnosis procedure (*compression ultrasonography*). We find that while the existing PubmedBERT tends to simply predict a choice that contains a word appearing in the question ("blood" for choice D), BioLinkBERT correctly predicts the answer (B). Our intuition is that citation links bring relevant documents and concepts together in the same context in pretraining (right),[5] which readily provides the multi-hop knowledge needed for the reasoning (center). Combined with the analysis on HotpotQA (§3.4.4), our results suggest that pretraining with document links consistently helps for multi-hop reasoning across domains (e.g., general documents with hyperlinks and biomedical articles with citation links).

MMLU-professional medicine. Table 3.9 shows the performance. Despite having just 340M parameters, BioLinkBERT$_{\texttt{large}}$ achieves 50% accuracy on this QA task, significantly outperforming the largest general-domain LM or QA models such as GPT-3 175B params (39% accuracy) and UnifiedQA 11B params (43% accuracy). This result shows that with an effective pretraining approach, a small domain-specialized LM can outperform orders of magnitude larger language models on QA tasks.

[5] For instance, as in Figure 3.4 (right), Ansari et al. (2015) in PubMed mention that *pancreatic cancer can induce deep vein thrombosis in leg*, and it cites a paper in PubMed, Piovella et al. (2002), which mention that *deep vein thrombosis is tested by compression ultrasonography*. Placing these two documents in the same context yields the complete multi-hop knowledge needed to answer the question ("*pancreatic cancer*" → "*deep vein thrombosis*" → "*compression ultrasonography*").

3.6 Related work

Retrieval-augmented LMs. Several works (Lewis et al., 2020c; Karpukhin et al., 2020; Oguz et al., 2020; Xie et al., 2022) introduce a retrieval module for LMs, where given an anchor text (e.g. question), retrieved text is added to the same LM context to improve model inference (e.g. answer prediction). These works show the promise of placing related documents in the same LM context at inference time, but they do not study the effect of doing so in pretraining. Guu et al. (2020) pretrain an LM with a retriever that learns to retrieve text for answering masked tokens in the anchor text. In contrast, our focus is not on retrieval, but on pretraining a general-purpose LM that *internalizes* knowledge that spans across documents, which is orthogonal to the above works (e.g., our pretrained LM could be used to initialize the LM component of these works). Additionally, we focus on incorporating document links such as hyperlinks, which can offer salient knowledge that common lexical retrieval methods may not provide (Asai et al., 2020).

Pretrain LMs with related documents. Several concurrent works use multiple related documents to pretrain LMs. Caciularu et al. (2021) place documents (news articles) about the same topic into the same LM context, and Levine et al. (2021) place sentences of high lexical similarity into the same context. Our work provides a general method to incorporate document links into LM pretraining, where lexical or topical similarity can be one instance of document links, besides hyperlinks. We focus on hyperlinks in this work, because we find they can bring in salient knowledge that may not be obvious via lexical similarity, and yield a more performant LM (§3.4.5). Additionally, we propose the DRP objective, which improves modeling multiple documents and relations between them in LMs (§3.4.5).

Hyperlinks and citation links for NLP. Hyperlinks are often used to learn better retrieval models. Chang et al. (2020); Asai et al. (2020); Seonwoo et al. (2021) use Wikipedia hyperlinks to train retrievers for open-domain question answering. Ma et al. (2021) study various hyperlink-aware pretraining tasks for retrieval. While these works use hyperlinks to learn retrievers, we focus on using hyperlinks to create better context for learning general-purpose LMs. Separately, Calixto et al. (2021) use Wikipedia hyperlinks to learn multilingual LMs. Citation links are often used to improve summarization and recommendation of academic papers (Qazvinian and Radev, 2008; Yasunaga et al., 2019; Bhagavatula et al., 2018; Khadka et al., 2020; Cohan et al., 2020). Here we leverage citation networks to improve pretraining general-purpose LMs.

Graph-augmented LMs. Several works augment LMs with graphs, typically, knowledge graphs (KGs) where the nodes capture entities and edges their relations. Zhang et al. (2019); He et al. (2020); Wang et al. (2021b) combine LM training with KG embeddings. Sun et al. (2020); Yasunaga et al. (2021); Zhang et al. (2022b) combine LMs and graph neural networks (GNNs) to jointly train

on text and KGs. Different from KGs, we use document graphs to learn knowledge that spans across documents.

3.7 Conclusion

We presented LinkBERT, a new language model (LM) pretraining method that incorporates document link knowledge such as hyperlinks. In both the general domain (pretrained on Wikipedia with hyperlinks) and biomedical domain (pretrained on PubMed with citation links), LinkBERT outperforms previous BERT models across a wide range of downstream tasks. The gains are notably large for multi-hop reasoning, multi-document understanding and few-shot question answering, suggesting that LinkBERT effectively internalizes salient knowledge through document links. Our results suggest that LinkBERT can be a strong pretrained LM to be applied to various knowledge-intensive tasks.

3.8 Supplementary

We apply the following fine-tuning hyperparameters to all models, including the baselines.

MRQA. For all the extractive question answering datasets, we use `max_seq_length` = 384 and a sliding window of size 128 if the lengths are longer than `max_seq_length`.

For the -tiny scale ($\text{BERT}_{\texttt{tiny}}$, $\text{LinkBERT}_{\texttt{tiny}}$), we choose learning rates from {5e-5, 1e-4, 3e-4}, batch sizes from {16, 32, 64}, and fine-tuning epochs from {5, 10}.

For -base ($\text{BERT}_{\texttt{base}}$, $\text{LinkBERT}_{\texttt{base}}$), we choose learning rates from {2e-5, 3e-5}, batch sizes from {12, 24}, and fine-tuning epochs from {2, 4}.

For -large ($\text{BERT}_{\texttt{large}}$, $\text{LinkBERT}_{\texttt{large}}$), we choose learning rates from {1e-5, 2e-5}, batch sizes from {16, 32}, and fine-tuning epochs from {2, 4}.

GLUE. We use `max_seq_length` = 128.

For the -tiny scale ($\text{BERT}_{\texttt{tiny}}$, $\text{LinkBERT}_{\texttt{tiny}}$), we choose learning rates from {5e-5, 1e-4, 3e-4}, batch sizes from {16, 32, 64}, and fine-tuning epochs from {5, 10}.

For -base and -large ($\text{BERT}_{\texttt{base}}$, $\text{LinkBERT}_{\texttt{base}}$, $\text{BERT}_{\texttt{large}}$, $\text{LinkBERT}_{\texttt{large}}$), we choose learning rates from {5e-6, 1e-5, 2e-5, 3e-5, 5e-5}, batch sizes from {16, 32, 64} and fine-tuning epochs from 3–10.

BLURB. We use `max_seq_length` = 512 and choose learning rates from {1e-5, 2e-5, 3e-5, 5e-5, 6e-5}, batch sizes from {16, 32, 64} and fine-tuning epochs from 1–120.

MedQA-USMLE. We use `max_seq_length` = 512 and choose learning rates from {1e-5, 2e-5, 3e-5}, batch sizes from {16, 32, 64} and fine-tuning epochs from 1–6.

Chapter 4

Fusing Structured Knowledge

In addition to the textual data discussed in the previous section, structured knowledge graphs offer additional information and domain-specific knowledge, complementing the textual knowledge. In this chapter, we introduce methods that enable language models to use both knowledge graph information and textual information.

4.1 Introduction

Pretraining learns self-supervised representations from massive raw data to help various downstream tasks (Bommasani et al., 2021). Language models (LMs) pretrained on large amounts of text data, such as BERT (Devlin et al., 2019) and GPTs (Brown et al., 2020), have shown strong performance on many natural language processing (NLP) tasks. The success of these models comes from deeply interactive (contextualized) representations of input tokens learned at scale via self-supervision (Devlin et al., 2019; Peters et al., 2018). Meanwhile, large knowledge graphs (KGs), such as Freebase (Bollacker et al., 2008), Wikidata (Vrandečić and Krötzsch, 2014) and ConceptNet (Speer et al., 2017), can provide complementary information to text data. KGs offer structured background knowledge by representing entities as nodes and relations between them as edges, and also offer scaffolds for structured, multi-step reasoning about entities (Yasunaga et al., 2021; Zhang et al., 2022b; Ren et al., 2020; 2021) (§4.4.5). The dual strengths of text data and KGs motivate research in pretraining deeply interactive representations of the two modalities at scale.

How to effectively combine text and KGs for pretraining is an open problem and presents challenges. Given text and KG, we need both (i) a *deeply bidirectional* model for the two modalities to interact, and (ii) a *self-supervised* objective to learn joint reasoning over text and KG at scale. Several existing works (Zhang et al., 2019; Xiong et al., 2020; Wang et al., 2021b; Agarwal et al., 2021; Sun et al., 2021) propose methods for self-supervised pretraining, but they fuse text and KG in a shallow or uni-directional manner. Another line of work (Yasunaga et al., 2021; Zhang et al., 2022b)

proposes bidirectional models for text and KG, but these models focus on finetuning on labeled downstream tasks and do not perform self-supervised learning. Consequently, existing methods may have limited their potential to model and learn deep interactions over text and KG.

To address both of the above challenges and fully unify the strengths of text and KG, we propose **DRAGON** (**D**eep Bidi**r**ectional L**a**n**g**uage-Kn**o**wledge Graph Pretrai**n**ing), an approach that performs deeply bidirectional, self-supervised pretraining of a language-knowledge model from text and KG. DRAGON has two core components: a cross-modal model that bidirectionally fuses text and KG, and a bidirectional self-supervised objective that learns joint reasoning over text and KG. Concretely, as in Figure 4.1, we take a text corpus and a KG as raw data, and create inputs for the model by sampling a text segment from the corpus and extracting a relevant subgraph from the KG via entity linking, obtaining a (*text*, *local KG*) pair. We use a cross-modal model to encode this input into fused representations, where each layer of the model encodes the text with an LM and the KG with a graph neural network (GNN), and fuses the two with a bidirectional modality interaction module (GreaseLM; Zhang et al. 2022b). We pretrain this model by unifying two self-supervised reasoning tasks: (1) masked language modeling (MLM), which masks and predicts tokens in the input text, and (2) link prediction, which drops and predicts edges in the input KG. The intuition is that by combining the two tasks, MLM makes the model use the text jointly with structured knowledge in the KG to reason about masked tokens in the text (e.g., in Figure 4.1, using the "round brush"–"art supply" multi-hop path from the KG helps), and link prediction makes the model use the KG structure jointly with the textual context to reason about missing links in the KG (e.g., recognizing that "round brush could be used for hair" from the text helps). This joint objective thus enables text to be grounded by KG structure and KG to be contextualized by text simultaneously, producing a deeply-unified language-knowledge pretrained model where information flows bidirectionally between text and KG for reasoning.

We pretrain DRAGON in two domains: a general domain, using the Book corpus and ConceptNet KG (Speer et al., 2017) (§4.4), and a biomedical domain, using the PubMed corpus and UMLS KG (Bodenreider, 2004) (§4.5). We show that DRAGON improves on existing LM and LM+KG models on diverse downstream tasks across domains. For the general domain, DRAGON outperforms RoBERTa (Liu et al., 2019), our base LM without KGs, on various commonsense reasoning tasks such as CSQA, OBQA, RiddleSense and HellaSwag, with +8% absolute accuracy gain on average. For the biomedical domain, DRAGON improves on the previous best LM, BioLinkBERT (Yasunaga et al., 2022b), and sets a new state of the art on BioNLP tasks such as MedQA and PubMedQA, with +3% accuracy gain. In particular, DRAGON exhibits notable improvements on QA tasks involving complex reasoning (+10% gain on multi-step, negation, hedge, or long context reasoning) and on downstream tasks with limited training data (+8% gain). These results show that our deep bidirectional self-supervision over text and KG produces significantly improved language-knowledge representations compared to existing models.

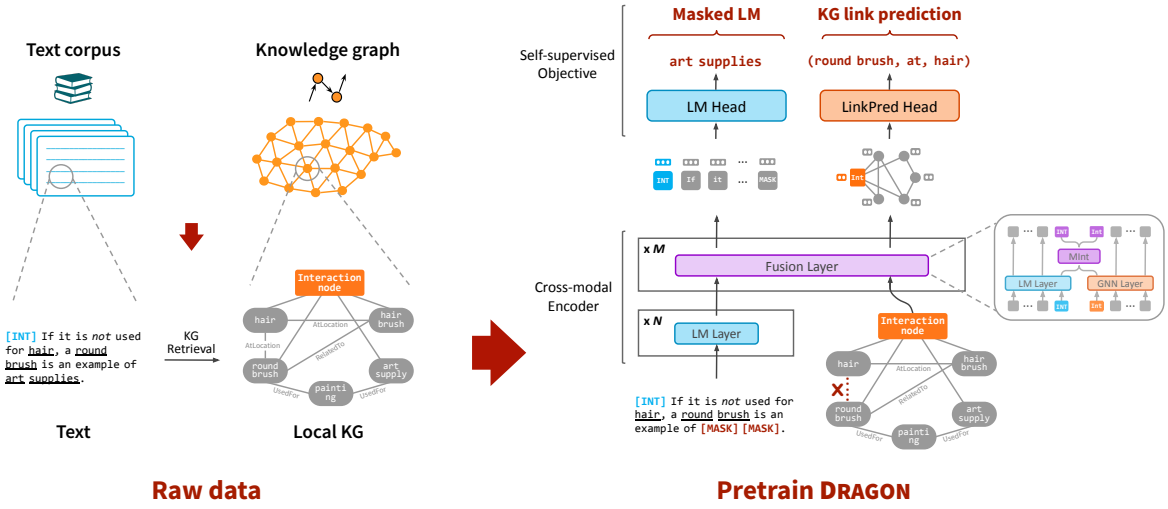


Figure 4.1: **Overview of our approach, DRAGON**. ***Left***: Given raw data of a text corpus and a large knowledge graph, we create aligned (text, local KG) pairs by sampling a text segment from the corpus and extracting a relevant subgraph from the KG (§4.3.1). As the structured knowledge in KG can ground the text and the text can provide the KG with rich context for reasoning, we aim to pretrain a language-knowledge model jointly from the text-KG pairs (DRAGON). ***Right***: To model the interactions over text and KG, DRAGON uses a cross-modal encoder that bidirectionally exchanges information between them to produce fused text token and KG node representations (§4.3.2). To pretrain DRAGON jointly on text and KG, we unify two self-supervised reasoning tasks: (1) masked language modeling, which masks some tokens in the input text and then predicts them, and (2) link prediction, which holds out some edges from the input KG and then predicts them. This joint objective encourages text and KG to mutually inform each other, facilitating the model to learn joint reasoning over text and KG (§4.3.3).

4.2 Related work

Knowledge-augmented LM pretraining. Knowledge integration is active research for improving LMs. One line of works is retrieval-augmented LMs (Guu et al., 2020; Lewis et al., 2020c; Borgeaud et al., 2022), which retrieve relevant text from a corpus and integrate it into LMs as additional knowledge. Orthogonal to these works, we focus on using knowledge bases as background knowledge, to ground reasoning about entities and facts.

Closest to our work are works that integrate knowledge bases in LM pretraining. One line of research aims to add entity features to LMs (Zhang et al., 2019; Peters et al., 2019; Rosset et al., 2020); Some works use the KG entity information or structure to create additional training signals (Xiong et al., 2020; Shen et al., 2020; Wang et al., 2021b; Liu et al., 2021a; Yu et al., 2022a; Ke et al., 2021); Several works add KG triplet information directly to the LM input (Liu et al., 2020; Sun et al., 2021; Agarwal et al., 2021; Sun et al., 2020; He et al., 2020). While these methods have achieved substantial progress, they typically propagate information between text and KG in a shallow or uni-directional (e.g., KG to text) manner, which might limit the potential to perform fully joint reasoning over the two modalities. To improve on the above works, we propose to bidirectionally interact text and KG via a deep cross-modal model and joint self-supervision, so that text and KG are grounded and contextualized by each other. We find that this improves model performance on various reasoning tasks (§4.4). Another distinction is that existing works in this space typically focus on adding entity- or triplet-level knowledge from KGs to LMs, and focus on solving entity/relation classification tasks. Our work significantly expands this scope in that we use larger KG subgraphs (200 nodes) as input to enable richer contextualization between KG and text, and we achieve performance improvements on a broader set of NLP tasks including QA, reasoning and text classification tasks.

KG-augmented question answering. Various works designed KG-augmented reasoning models for question answering (Lin et al., 2019; Feng et al., 2020; Lv et al., 2020; Wang et al., 2022a; Mihaylov and Frank, 2018; Yang et al., 2019; Sun et al., 2018; 2019a; Yan et al., 2021; Sun et al., 2022; Xu et al., 2022). In particular, recent works such as QAGNN (Yasunaga et al., 2021) and GreaseLM (Zhang et al., 2022b) suggest that a KG can scaffold reasoning about entities with its graph structure, and help for complex question answering (e.g., negation, multi-hop reasoning). These works typically focus on training or finetuning models on particular QA datasets. In contrast, we generalize this and integrate KG-augmented reasoning into general-purpose pretraining. Our motivation is that self-supervised pretraining allows the model to learn from larger and more diverse data, helping to learn richer interactions between text and KGs and to acquire more diverse reasoning abilities beyond specific QA tasks. We find that our proposed pretraining approach (DRAGON) offers significant boosts over the baseline QA models (e.g. GreaseLM) on diverse downstream tasks (§4.4). This opens a new research avenue in scaling up various carefully-designed QA models to pretraining.

KG representation learning. Our link prediction task used in pretraining is motivated by research in KG representation learning. Link prediction is a fundamental task in KGs (Trouillon et al., 2016; Kazemi and Poole, 2018), and various works study methods to learn KG entity and relation embeddings for link prediction, such as TransE (Bordes et al., 2013), DistMult (Yang et al., 2015) and RotatE (Sun et al., 2019b). Several works additionally use textual data or pretrained LMs to help learn KG embeddings and link prediction (Riedel et al., 2013; Toutanova et al., 2015; Xie et al., 2016; Yao et al., 2019; Kim et al., 2020; Li et al., 2022a). While these works focus on the KG-side representations, we extend the scope and use the KG-side objective (link prediction) jointly with a text-side objective (language modeling) to train a mutually-interactive text-KG model.

4.3 Approach

We propose Deep Bidirectional Language-Knowledge Graph Pretraining (DRAGON), an approach that performs deeply bidirectional, self-supervised pretraining of a language-knowledge model from text and KG. Specifically, as illustrated in Figure 4.1, we take a text corpus and a large knowledge graph as raw data, and create input instances for the model by sampling coarsely-aligned (text segment, local KG) pairs (§4.3.1). To learn mutual interactions over text and KG, DRAGON consists of a cross-modal encoder (GreaseLM) that fuses the input text-KG pair bidirectionally (§4.3.2), and a pretraining objective that performs bidirectional self-supervision on the text-KG input (§4.3.3). Our pretraining objective unifies masked language modeling (MLM) and KG link prediction (LinkPred) to make text and KG mutually inform each other and learn joint reasoning over them. Finally, we describe how we finetune the pretrained DRAGON model for downstream tasks (§4.3.4).

Definitions. We define a text corpus $\mathcal{W}$ as a set of text segments $\mathcal{W} = \{W\}$, and each text segment W as a sequence of tokens (words), $W = (w_1, ..., w_I)$. We define a knowledge graph (KG) as a multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of entity nodes in the KG and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ is the set of edges (triplets) that connect nodes in $\mathcal{V}$, with $\mathcal{R}$ being the set of relation types $\{r\}$. Each triplet (h, r, t) in a KG can represent a knowledge fact such as $(\texttt{Paris}, \texttt{in}, \texttt{France})$. As a raw KG is often large, with millions of nodes, a subgraph of the raw KG (*local KG*) is considered: $G = (V, E)$ where $V = \{v_1, ..., v_J\} \subseteq \mathcal{V}$ and $E \subseteq \mathcal{E}$. We define a language-knowledge model to be a composition of two functions, $f_{\text{head}}(f_{\text{enc}}(X))$, where the encoder f_{enc} takes in an input $X =$ (text segment W, local KG G), and produces a contextualized vector representation for each text token, $(\mathbf{H}_1, ..., \mathbf{H}_I)$, and for each KG node, $(\mathbf{V}_1, ..., \mathbf{V}_J)$. A language model is a special case of a language-knowledge model with no KG ($J = 0$). The head f_{head} uses these representations to perform self-supervised tasks in the pretraining step and to perform downstream tasks in the finetuning step.

4.3.1 Input representation

Given a text corpus $\mathcal{W}$ and a large knowledge graph $\mathcal{G}$, we create input instances for the model by preparing (text segment W, local KG G) pairs. We want each pair's text and KG to be (roughly) semantically aligned so that the text and KG can mutually inform each other and facilitate the model to learn interactive reasoning between the two modalities. Specifically, for each text segment W from $\mathcal{W}$, we extract a relevant local KG G for it from $\mathcal{G}$ via the following KG retrieval process.

KG retrieval. Given a text segment W, we link entity mentions in W to entity nodes in $\mathcal{G}$ to get an initial set of nodes V_{el}. We then add their 2-hop bridge nodes from $\mathcal{G}$ to get the total retrieved nodes $V \subseteq \mathcal{V}$. Lastly, we add all edges that span these nodes in $\mathcal{G}$ to get $E \subseteq \mathcal{E}$, which yields the final local KG, $G = (V, E)$, as well as our final input instance $X = (W, G)$. §4.8.1 provides more details on KG retrieval. Henceforth, we use "KG" to refer to this local KG G unless noted otherwise.

Modality interaction token/node. For each resulting (text, KG) pair, we further add a special token (interaction token) w_{int} to the text and a special node (interaction node) v_{int} to the KG, which will serve as an information pooling point for each modality as well as an interface for modality interaction in our cross-modal encoder (§4.3.2). Specifically, we prepend w_{int} to the original text $W = (w_1, ..., w_I)$, and connect v_{int} to the entity-linked nodes in the original KG, $V_{\text{el}} \subseteq V = \{v_1, ..., v_J\}$, using a new relation type r_{el}. The interaction token and node can also be used to produce a pooled representation of the whole input, e.g., when finetuning for classification tasks (§4.3.4).

4.3.2 Cross-modal encoder

To model mutual interactions over the text and KG, we use a bidirectional sequence-graph encoder for f_{enc} which takes in the text tokens and KG nodes and exchanges information across them for multiple layers to produce a fused representation of each token and node (Figure 4.1 right):

$$(\mathbf{H}_{\text{int}}, \mathbf{H}_1, ..., \mathbf{H}_I), (\mathbf{V}_{\text{int}}, \mathbf{V}_1, ..., \mathbf{V}_J) = f_{\text{enc}}((w_{\text{int}}, w_1, ..., w_I), (v_{\text{int}}, v_1, ..., v_J)) \tag{4.1}$$

While we may use any deep bidirectional sequence-graph encoder for f_{enc}, for controlled comparison with existing works, we adopt the current top-performing sequence-graph architecture, GreaseLM (Zhang et al., 2022b), which combines Transformers (Vaswani et al., 2017) and graph neural networks (GNNs) to fuse text-KG inputs.

Specifically, GreaseLM first uses N layers of Transformer language model (LM) layers to map the input text into initial token representations, and uses KG node embeddings to map the input KG nodes into initial node representations,

$$(\mathbf{H}_{\text{int}}^{(0)}, \mathbf{H}_1^{(0)}, ..., \mathbf{H}_I^{(0)}) = \text{LM-Layers}(w_{\text{int}}, w_1 ..., w_I), \tag{4.2}$$

$$(\mathbf{V}_{\text{int}}^{(0)}, \mathbf{V}_1^{(0)}, ..., \mathbf{V}_J^{(0)}) = \text{Node-Embedding}(v_{\text{int}}, v_1, ..., v_J). \tag{4.3}$$

Then it uses M layers of text-KG fusion layers to encode these token/node representations jointly into the final token/node representations,

$$(\mathbf{H}_{\text{int}}, ..., \mathbf{H}_I), (\mathbf{V}_{\text{int}}, ..., \mathbf{V}_J) = \text{Fusion-Layers}((\mathbf{H}_{\text{int}}^{(0)}, ..., \mathbf{H}_I^{(0)}), (\mathbf{V}_{\text{int}}^{(0)}, ..., \mathbf{V}_J^{(0)})), \tag{4.4}$$

where each of the fusion layers (ℓ=1, ..., M) performs the following:

$$(\widetilde{\mathbf{H}}_{\text{int}}^{(\ell)}, \mathbf{H}_1^{(\ell)}, ..., \mathbf{H}_I^{(\ell)}) = \text{LM-Layer}(\mathbf{H}_{\text{int}}^{(\ell-1)}, \mathbf{H}_1^{(\ell-1)}, ..., \mathbf{H}_I^{(\ell-1)}), \tag{4.5}$$

$$(\widetilde{\mathbf{V}}_{\text{int}}^{(\ell)}, \mathbf{V}_1^{(\ell)}, ..., \mathbf{V}_J^{(\ell)}) = \text{GNN-Layer}(\mathbf{V}_{\text{int}}^{(\ell-1)}, \mathbf{V}_1^{(\ell-1)}, ..., \mathbf{V}_J^{(\ell-1)}), \tag{4.6}$$

$$[\mathbf{H}_{\text{int}}^{(\ell)}; \mathbf{V}_{\text{int}}^{(\ell)}] = \text{MInt}([\widetilde{\mathbf{H}}_{\text{int}}^{(\ell)}; \widetilde{\mathbf{V}}_{\text{int}}^{(\ell)}]). \tag{4.7}$$

Here GNN induces graph structure-aware representations of KG nodes, $[\cdot;\cdot]$ does concatenation, and MInt (modality interaction module) exchanges information between the interaction token (text side) and interaction node (KG side) via an MLP. For more details on GreaseLM, we refer readers to (Zhang et al., 2022b).

4.3.3 Pretraining objective

We aim to pretrain the DRAGON model so that it learns joint reasoning over text and a KG. To ensure that the text and KG mutually inform each other and the model learns bidirectional information flow, we unify two self-supervised reasoning tasks: masked language modeling and KG link prediction.

Masked language modeling (MLM). MLM is a common pretraining task used for language models (e.g., BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019)), which masks some tokens in the input text and predicts them. This task makes the model use non-masked context to reason about masked tokens, and in particular, as our approach takes a joint text-KG pair as input, we expect that MLM can encourage the model to learn to use the text *jointly with* structured knowledge in the KG to reason about masks in the text (e.g., in the example of Figure 4.1, besides the textual context, recognizing the "round brush"–"art supply" path from the KG can help together to predict the masked tokens "art supplies").

Concretely, to perform the MLM task, we mask a subset of tokens in the input text, $M \subseteq W$, with a special token `[MASK]`, and let the task head f_{head} be a linear layer that takes the contextualized token vectors $\{\mathbf{H}_i\}$ from the encoder to predict the original tokens. The objective is a cross-entropy loss:

$$\mathcal{L}_{\text{MLM}} = -\sum_{i \in M} \log p(w_i \mid \mathbf{H}_i). \tag{4.8}$$

Link prediction (LinkPred). While the MLM task predicts for the text side, link prediction holds out some edges and predicts them for the input KG. Link prediction is a fundamental task in KGs (Sun et al., 2019b) and makes the model use the structure of KGs to perform reasoning (e.g., using a compositional path "X's mother's husband is Y" to deduce a missing link "X's father is Y"). In particular, as our approach takes a joint text-KG pair as input, we expect that link prediction can encourage the model to learn to use the KG structure *jointly with* the textual context to reason about missing links in the KG (e.g., in Figure 4.1, besides the KG structure, recognizing that "round brush could be used for hair" from the text can help together to predict the held-out edge `(round_brush, at, hair)`).

Concretely, to perform the link prediction task, we hold out a subset of edge triplets from the input KG, $S = \{(h, r, t)\} \subseteq E$. For the task head f_{head}, we adopt a KG representation learning framework, which maps each entity node (h or t) and relation (r) in the KG to a vector, $\mathbf{h}, \mathbf{t}, \mathbf{r}$, and defines a scoring function $\phi_r(\mathbf{h}, \mathbf{t})$ to model positive/negative triplets. Specifically, we let $\mathbf{h} = \mathbf{V}_h$, $\mathbf{t} = \mathbf{V}_t$, $\mathbf{r} = \mathbf{R}_r$, with $\{\mathbf{V}_j\}$ being the contextualized node vectors from the encoder, and $\mathbf{R} = \{\mathbf{r}_1, ..., \mathbf{r}_{|\mathcal{R}|}\}$ being learnable relation embeddings. We consider a KG triplet scoring function $\phi_r(\mathbf{h}, \mathbf{t})$ such as

$$\text{DistMult (Yang et al., 2015)}: \langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle, \tag{4.9}$$

$$\text{TransE (Bordes et al., 2013)}: -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|, \tag{4.10}$$

$$\text{RotatE (Sun et al., 2019b)}: -\|\mathbf{h} \odot \mathbf{r} - \mathbf{t}\|, \tag{4.11}$$

where $\langle \cdot, \cdot, \cdot \rangle$ denotes the trilinear dot product and $\odot$ the Hadamard product. A higher ϕ indicates a higher chance of (h, r, t) being a positive triplet (edge) instead of negative (no edge). We analyze the choices of scoring functions in §4.4.7. For training, we optimize the objective:

$$\mathcal{L}_{\text{LinkPred}} = \sum_{(h,r,t) \in S} \left(-\log \sigma(\phi_r(\mathbf{h}, \mathbf{t}) + \gamma) + \frac{1}{n} \sum_{(h',r,t')} \log \sigma(\phi_r(\mathbf{h}', \mathbf{t}') + \gamma) \right), \tag{4.12}$$

where (h', r, t') are n negative samples corresponding to the positive triplet (h, r, t), γ is the margin, and σ is the sigmoid function. The intuition of this objective is to make the model predict triplets of the held-out edges S as positive and other random triplets as negative.

Joint training. To pretrain DRAGON, we optimize the MLM and LinkPred objectives jointly: $\mathcal{L} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{LinkPred}}$. This joint objective unifies the effects of MLM and LinkPred, which encourage the model to simultaneously ground text with KG structure and contextualize KG with text,

facilitating bidirectional information flow between text and KGs for reasoning. We show in §4.4.7 that the joint objective yields a more performant model than using one of the objectives alone.

4.3.4 Finetuning

Lastly, we describe how we finetune DRAGON for downstream tasks such as text classification and multiple-choice QA (MCQA). Given an input text W (e.g., concatenation of a question and an answer choice in the case of MCQA), we follow the same steps as §4.3.1 and §4.3.2 to retrieve a relevant local KG G and encode them jointly into contextualized token/node vectors, $(\mathbf{H}_{\text{int}}, \mathbf{H}_1, ..., \mathbf{H}_I)$, $(\mathbf{V}_{\text{int}}, \mathbf{V}_1, ..., \mathbf{V}_J)$. We then compute a pooled representation of the whole input as $\mathbf{X} = \text{MLP}(\mathbf{H}_{\text{int}}, \mathbf{V}_{\text{int}}, \mathbf{G})$, where $\mathbf{G}$ denotes attention-based pooling of $\{\mathbf{V}_j \mid v_j \in \{v_1, ..., v_J\}\}$ using $\mathbf{H}_{\text{int}}$ as a query. Finally, the pooled representation $\mathbf{X}$ is used to perform the downstream task, in the same way as how the `[CLS]` representation is used in LMs such as BERT and RoBERTa.

The difference from the original GreaseLM work is that while GreaseLM only performs finetuning as described in this section (hence, it is an LM *finetuned* with KGs), DRAGON performs self-supervised pretraining as described in §4.3.3 (hence, it can be viewed as an LM *pretrained + finetuned* with KGs).

4.4 Experiments: General domain

We experiment with the proposed approach DRAGON in a general domain first. We pretrain DRAGON using the Book corpus and ConceptNet KG (§4.4.1), and evaluate on diverse downstream tasks (§4.4.2). We show that DRAGON significantly improves on existing models (§4.4.4). We extensively analyze the effect of DRAGON's key design choices such as self-supervision and use of KGs (§4.4.5, 4.4.6, 4.4.7). We also experiment in the biomedical domain in §4.5.

4.4.1 Pretraining setup

Data. For the text data, we use BookCorpus (Zhu et al., 2015), a general-domain corpus widely used in LM pretraining (e.g., BERT, RoBERTa). It has 6GB of text from online books. For the KG data, we use ConceptNet (Speer et al., 2017), a general-domain knowledge graph designed to capture background commonsense knowledge. It has 800K nodes and 2M edges in total. To create a training instance, we sample a text segment of length up to 512 tokens from the text corpus, then retrieve a relevant KG subgraph of size up to 200 nodes (details in §4.8.1), by which we obtain an aligned (text, local KG) pair.

Implementation. For our encoder (§4.3.2), we use the GreaseLM architecture (Zhang et al., 2022b) (19 LM layers followed by 5 text-KG fusion layers; 360M parameters in total). As done by (Zhang et al., 2022b), we initialize parameters in the LM component with the RoBERTa-Large

release (Liu et al., 2019) and initialize the KG node embeddings with pre-computed ConceptNet entity embeddings (details in §4.8.1). For the link prediction objective (§4.3.3, Equation 4.12), we use DistMult (Yang et al., 2015) for KG triplet scoring, with a negative exampling of 128 triplets and a margin of $\gamma = 0$. To pretrain the model, we perform MLM with a token masking rate of 15% and link prediction with an edge drop rate of 15%. We pretrain for 20,000 steps with a batch size of 8,192 and a learning rate of 2e-5 for parameters in the LM component and 3e-4 for the others. Training took 7 days on eight A100 GPUs using FP16. Additional details on the hyperparameters can be found in §4.8.1.

4.4.2 Downstream evaluation tasks

We finetune and evaluate DRAGON on nine diverse commonsense reasoning benchmarks: CommonsenseQA (**CSQA**) (Talmor et al., 2019), OpenbookQA (**OBQA**) (Mihaylov et al., 2018), RiddleSense (**Riddle**) (Lin et al., 2021), AI2 Reasoning Challenge–Challenge Set (**ARC**) (Clark et al., 2018), **CosmosQA** (Huang et al., 2019b), **HellaSwag** (Zellers et al., 2019), Physical Interaction QA (**PIQA**) (Bisk et al., 2020), Social Interaction QA (**SIQA**) (Sap et al., 2019), and Abductive Natural Language Inference (**aNLI**) (Bhagavatula et al., 2020). For CSQA, we follow the in-house data splits used by prior works (Lin et al., 2019). For OBQA, we follow the original setting where the models only use the question as input and do not use the extra science facts. §4.8.2 provides the full details on these tasks and data splits. Hyperparameters used for finetuning can be found in §4.8.1.

4.4.3 Baselines

LM. To study the effect of using KGs, we compare DRAGON with the vanilla language model, RoBERTa (Liu et al., 2019). As we initialize DRAGON's parameters using the RoBERTa-Large release (§4.4.1), for fair comparison, we let the baseline be such that we take the RoBERTa-Large release and continue pretraining it with the vanilla MLM objective on the same text data for the same number of steps as DRAGON. Hence, the only difference is that DRAGON uses KGs during pretraining while RoBERTa does not. We then perform standard LM finetuning of RoBERTa on downstream tasks.

LM finetuned with KG. We also compare with existing KG-augmented QA models, QAGNN (Yasunaga et al., 2021) and GreaseLM (Zhang et al., 2022b), which *finetune* a vanilla LM (i.e. RoBERTa-Large) with a KG on downstream tasks, but do not *pretrain* with a KG. GreaseLM is the existing top-performing model in this paradigm. We use the same encoder architecture as GreaseLM for DRAGON; the difference from the original GreaseLM work is that DRAGON performs self-supervised pretraining while GreaseLM did not.

4.4.4 Results

Table 4.1 shows performance on the 9 downstream commonsense reasoning tasks. Across all tasks, DRAGON consistently outperforms the existing LM (RoBERTa) and KG-augmented QA models (QAGNN, GreaseLM), e.g., +7% absolute accuracy boost over RoBERTa and +5% over GreaseLM on *OBQA*. These accuracy boosts indicate the advantage of DRAGON over RoBERTa (KG reasoning) and over GreaseLM (pretraining). The gain is especially significant on datasets that have small training data such as *ARC*, *Riddle* and *OBQA*, and datasets that require complex reasoning such as *CosmosQA* and *HellaSwag*, which we analyze in more detail in the following sections.

4.4.5 Analysis: Effect of knowledge graph

The first key contribution of DRAGON (w.r.t. existing LM pretraining methods) is that we incorporate KGs. We find that this significantly improves the model's performance for robust and complex reasoning, such as resolving multi-step reasoning and negation, as we discuss below.

Quantitative analysis. In Table 4.2, we study downstream task performance of DRAGON on questions involving complex reasoning. Building on (Yasunaga et al., 2021; Zhang et al., 2022b), we consider several proxies to categorize complex questions: (i) presence of negation (e.g. *no*, *never*), (ii) presence of conjunction (e.g. *and*, *but*), (iii) presence of hedge (e.g. *sometimes*, *maybe*), (iv) number of prepositional phrases, and (v) number of entity mentions. Having negation or conjunction indicates logical multi-step reasoning, having more prepositional phrases or entity mentions indicates involving more reasoning steps or constraints, and having hedge terms indicates involving complex textual nuance. DRAGON significantly outperforms the baseline LM (RoBERTa) across all these categories (e.g., +14% accuracy for negation), which confirms that our joint language-knowledge pretraining boosts reasoning performance. DRAGON also consistently outperforms the existing KG-augmented QA models (QAGNN, GreaseLM). We find that QAGNN and GreaseLM only improve moderately on RoBERTa for some categories like conjunction or many prepositional phrases ($=2, 3$), but DRAGON provides substantial boosts. This suggests that through self-supervised pretraining with larger and diverse data, DRAGON has learned more general-purpose reasoning abilities than the finetuning-only models like GreaseLM.

	CSQA	OBQA	Riddle	ARC	CosmosQA	HellaSwag	PIQA	SIQA	aNLI
RoBERTa (Liu et al., 2019)	68.7	64.9	60.7	43.0	80.5	82.3	79.4	75.9	82.7
QAGNN (Yasunaga et al., 2021)	73.4	67.8	67.0	44.4	80.7	82.6	79.6	75.7	83.0
GreaseLM (Zhang et al., 2022b)	74.2	66.9	67.2	44.7	80.6	82.8	79.6	75.5	83.3
DRAGON (**Ours**)	**76.0**	**72.0**	**71.3**	**48.6**	**82.3**	**85.2**	**81.1**	**76.8**	**84.0**

Table 4.1: Accuracy on downstream commonsense reasoning tasks. DRAGON consistently outperforms the existing LM (RoBERTa) and KG-augmented QA models (QAGNN, GreaseLM) on all tasks. The gain is especially significant on tasks that have small training data (*OBQA*, *Riddle*, *ARC*) and tasks that require complex reasoning (*CosmosQA*, *HellaSwag*).

	Negation	**Conjunction**	**Hedge**	**# Prepositional Phrases**				**# Entities**
				0	1	2	3	¿10
RoBERTa	61.7	70.9	68.6	67.6	71.0	71.1	73.1	74.5
QAGNN	65.1	74.5	74.2	72.1	71.6	75.6	71.3	78.6
GreaseLM	65.1	74.9	76.6	75.6	73.8	74.7	73.6	79.4
DRAGON (**Ours**)	**75.2**	**79.6**	**77.5**	**79.1**	**78.2**	**77.8**	**80.9**	**83.5**

Table 4.2: Accuracy of DRAGON on *CSQA* + *OBQA* dev sets for **questions involving complex reasoning** such as negation terms, conjunction terms, hedge terms, prepositional phrases, and more entity mentions. DRAGON consistently outperforms the existing LM (RoBERTa) and KG-augmented QA models (QAGNN, GreaseLM) in these complex reasoning settings.

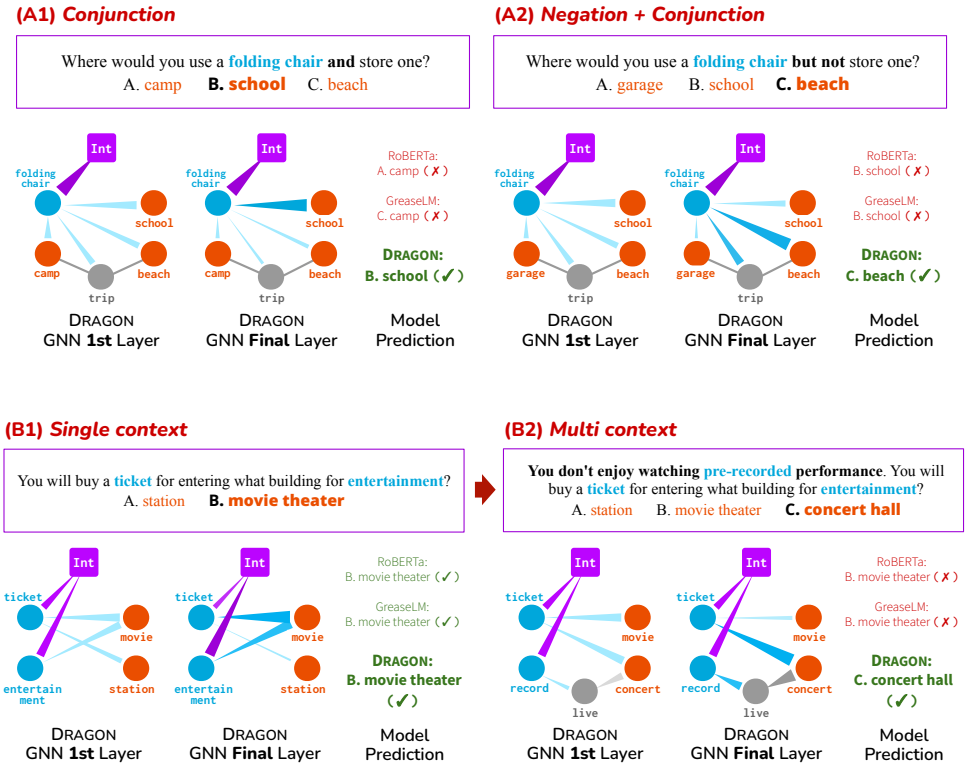


Figure 4.2: Analysis of DRAGON's graph reasoning, where we visualize how graph attention weights and final predictions change given question variations. Darker and thicker edges indicate higher attention weights. **DRAGON exhibits abilities to extrapolate and perform robust reasoning**. DRAGON adjusts the entity attention weights and final predictions accordingly when conjunction or negation is given about entities (A1, A2) or when extra context is added to an original question (B1→B2), but existing models like RoBERTa struggle to predict the correct answers. **A1:** DRAGON's final GNN layer shows strong attention to "school" but weak attention to "trip", likely because the question states "*and* store one"—hence, the chair is *not* used for a trip. **A2:** DRAGON shows strong attention to "trip" and "beach", likely because the question now states "*but not* store one"—hence, the chair *is* used for a trip. **B1→B2:** DRAGON's final GNN layer shows strong attention to "movie" in the original question (B1), but after adding the extra context "don't enjoy pre-record" (B2), DRAGON shows strong attention to "live" and "concert", leading to making the correctly adjusted prediction "concert hall". One interpretation of these findings is that DRAGON leverages the KG's graph structure as a scaffold for performing complex reasoning. This insight is related to recent works that provide LMs with scratch space for intermediate reasoning (Yasunaga et al., 2021; Nye et al., 2021; Wei et al., 2022).

Qualitative analysis. Using the *CSQA* dataset, we further conducted case studies on the behavior of DRAGON's KG reasoning component, where we visualize how graph attention weights change given different question variations (Figure 4.2). We find that DRAGON exhibits abilities to extrapolate and perform robust reasoning. For instance, DRAGON adjusts the entity attention weights and final predictions accordingly when we add conjunction or negation about entities (A1, A2) or when we add extra context to an original question (B1→B2), but existing models like RoBERTa struggle to predict the correct answers. As these questions are more complex than ones typically seen in the *CSQA* training set, our insight is that while vanilla LMs (RoBERTa) and finetuning (GreaseLM) have limitation in learning complex reasoning, KG-augmented pretraining (DRAGON) helps acquire generalizable reasoning abilities that extrapolate to harder test examples.

Method	CosmosQA (10% train)	PIQA (10% train)
RoBERTa	72.2	66.4
GreaseLM	73.0	67.0
DRAGON (**Ours**)	**77.9**	**72.3**

Table 4.3: Performance in low-resource setting where 10% of finetuning data is used. DRAGON attains large gains, suggesting its benefit for downstream data efficiency.

Method	CSQA	OBQA
GreaseLM	74.2	66.9
GreaseLM-Ex	73.9	66.2
DRAGON (**Ours**)	76.0	72.0
DRAGON-Ex (**Ours**)	**76.3**	**72.8**

Table 4.4: Downstream performance when model capacity—number of text-KG fusion layers—is increased ("-Ex"). Increased capacity does not help for the finetuning-only model (GreaseLM), but helps when pretrained (DRAGON), suggesting the promise of DRAGON to be further scaled up.

Ablation Type	Ablation	CSQA	OBQA
Pretraining objective	MLM + LinkPred (**final**)	**76.0**	**72.0**
	MLM only	74.3	67.2
	LinkPred only	73.8	66.4
LinkPred head	DistMult (**final**)	**76.0**	**72.0**
	TransE	75.7	71.4
	RotatE	75.8	71.7
Cross-modal model	Bidirectional interaction (**final**)	**76.0**	**72.0**
	Concatenate at end	74.5	68.0
KG structure	Use graph (**final**)	**76.0**	**72.0**
	Convert to sentence	74.7	70.1

Table 4.5: Ablation study of DRAGON. Using joint pretraining objective MLM + LinkPred (§4.3.3) outperforms using one of them only. All variants of LinkPred scoring models (DistMult, TransE, RotatE) outperform the baseline without LinkPred ("MLM only"), suggesting that DRAGON can be combined with various KG representation learning models. Cross-modal model with bidirectional modality interaction (§4.3.2) outperforms combining text and KG representations only at the end. Finally, using KG as graph outperforms converting KG as sentences, suggesting the benefit of graph structure for reasoning.

4.4.6 Analysis: Effect of pretraining

Another key contribution of DRAGON (w.r.t. existing QA models like GreaseLM) is pretraining. Here we discuss when and why our pretraining is useful. Considering the three core factors in machine learning (data, task complexity, and model capacity), pretraining helps when the available downstream task data is smaller compared to the downstream task complexity or model capacity. Concretely, we find that DRAGON is especially helpful for the following three scenarios.

Downstream tasks with limited data. In Table 4.1, we find that DRAGON provides significant boosts over GreaseLM on downstream tasks with limited finetuning data available, such as *ARC* (3K training instances; +4% accuracy gain), *Riddle* (3K instances; +4% accuracy) and *OBQA* (5K instances; +5% accuracy). For other tasks, we also experimented with a low-resource setting where 10% of finetuning data is used (Table 4.3). Here we also see that DRAGON attains significant gains over GreaseLM (+5% accuracy on *PIQA*), suggesting the improved data-efficiency of DRAGON.

Complex downstream tasks. In Table 4.1, we find that DRAGON provides substantial gains over GreaseLM on downstream tasks involving more complex reasoning, such as *CosmosQA* and *HellaSwag*, where the inputs have longer context and more entities (thus bigger local KGs). For these tasks, improvements of GreaesLM over RoBERTa were small (+0.1% on *CosmosQA*), but DRAGON provides substantial boosts (+1.8%). Our insight is that through self-supervised pretraining with larger and more diverse data, DRAGON has learned richer text-KG interactions than GreaseLM, enabling solving more complex downstream tasks. Similarly, as seen in §4.4.5, DRAGON also attains large gains over GreaseLM on complex questions containing negation, conjunction and prepositional phrases (Table 4.2), and extrapolates to questions more complex than seen in training sets (Figure 4.2).

Increased model capacity. In Table 4.4, we study downstream performance when the model capacity is increased—the number of text-KG fusion layers is increased from 5 to 7—for both GreaseLM and DRAGON. We find that increased capacity does not help for the finetuning-only model (GreaseLM) as was also reported in the original GreaseLM paper, but it helps when pretrained (DRAGON). This result reveals that increased model capacity can actually be beneficial when combined with pretraining, and suggests the promise of DRAGON to be further scaled up.

4.4.7 Analysis: Design choices of DRAGON

Pretraining objective (Table 4.5 top). The first important design choice of DRAGON is the joint pretraining objective: MLM + LinkPred (§4.3.3). Using the joint objective outperforms using MLM or LinkPred alone (+5% accuracy on *OBQA*). This suggests that having the bidirectional self-supervised tasks on text and KG facilitates the model to fuse the two modalities for reasoning.

Link prediction head choice (Table 4.5 middle 1). KG representation learning is an active area of research, and various KG triplet scoring models are proposed (Equation 4.9). We hence experimented with using different scoring models for DRAGON's link prediction head (§4.3.3). We find that while DistMult has a slight edge, all variants we tried (DistMult, TransE, RotatE) are effective, outperforming the baseline without LinkPred ("MLM only"). This result suggests the generality of DRAGON and its promise to be combined with various KG representation learning techniques.

Cross-modal model (Table 4.5 middle 2). Another core component of DRAGON is the cross-modal encoder with bidirectional text-KG fusion layers (§4.3.2). We find that if we ablate them and simply concatenate text and KG representations at the end, the performance drops substantially. This result suggests that deep bidirectional fusion is crucial to model interactions over text and KG for reasoning.

KG structure & GNN (Table 4.5 bottom). The final key design of DRAGON is that we leverage the graph structure of KGs via a sequence-graph encoder and link prediction objective. Here we experimented with an alternative pretraining method that drops the graph structure: we convert triplets in the local KG into sentences using a template (Feng et al., 2020), append them to the main text input, and perform vanilla MLM pretraining. We find that DRAGON substantially outperforms this variant (+2% accuracy on *OBQA*), which suggests that the graph structure of KGs helps the model perform reasoning.

4.4.8 Analysis: Why GNN is useful for question answering?

We delve deeper into why our GNN-based model excels in question answering and reasoning tasks, even handling complex ones involving negation and conjunction. Recent studies show the effectiveness of GNNs in modeling diverse graph algorithms (Xu et al., 2020), including tasks such as knowledge graph reasoning and executing logical queries on a KG (Gentner, 1983; Ren and Leskovec, 2020):

$$V_?.\ \exists V:\ \text{Located}(\text{Europe}, V) \\ \wedge \neg \text{Held}(\text{World Cup}, V) \wedge \text{President}(V, V_?)$$

*("Who are the presents of European countries that have **not** held the World Cup?")*

Viewing such logical queries as input "questions", we conducted a pilot study where we apply DRAGON to learn the task of executing logical queries on a KG—including complex queries that

contain negation or multi-hop relations about entities. In this task, we find that DRAGON significantly outperforms a baseline model that only uses an LM but not a GNN:

Methods	**Hit@3 on FB15k**
LM-only	15
DRAGON (**Ours**)	**40**

Table 4.6: Performance in learning to answer complex logical queries on a KG.

The result confirms that GNNs are indeed useful for modeling complex query answering. This provides an intuition that DRAGON can be useful for answering complex natural language questions too, which could be viewed as executing soft queries—natural language instead of logical—using a KG. From this "KG query execution" intuition, we may also draw an interpretation that the KG and GNN can provide a *scaffold* for the model to reason about entities mentioned in the question.

4.5 Experiments: Biomedical domain

Biomedicine is a domain with extensive background knowledge (Brown et al., 1999; Lipscomb, 2000; Zitnik et al., 2018; Bommasani et al., 2021), and experts curate various knowledge bases for it (Ashburner et al., 2000; Bodenreider, 2004; Wishart et al., 2018; Ruiz et al., 2021). We hypothesize that these biomedical KGs can enable deeper understanding and reasoning about biomedical text. With this motivation, we pretrain DRAGON on a biomedical corpus and KG, and evaluate on biomedical downstream tasks.

Pretraining setup. For the text data, we use PubMed (of Medicine, 1996), a widely-used corpus in biomedial LM training (e.g., BioBERT (Lee et al., 2020), PubmedBERT (Gu et al., 2021)). It contains the abstracts of biomedical papers on PubMed and has 21GB of text. For the KG data, we use the Unified Medical Language System (UMLS) (Bodenreider, 2004), a widely-used knowledge graph in biomedicine. It has 300K nodes and 1M edges in total.

For training, we follow the same procedure as the experiment in the general domain (§4.4.1), except that we initialize DRAGON's LM component with BioLinkBERT-Large (Yasunaga et al., 2022b), the state-of-the-art biomedical LM, instead of RoBERTa-Large. Note that while "BioLinkBERT" has "Link" in its name, it is not about KG links but about citation links that the model was originally pretrained with.

Method	**MedQA**	**PubMedQA**	**BioASQ**
BioBERT (Lee et al., 2020)	36.7	60.2	84.1
PubmedBERT (Gu et al., 2021)	38.1	55.8	87.5
BioLinkBERT (Yasunaga et al., 2022b)	44.6	72.2	94.8
+ QAGNN	45.0	72.1	95.0
+ GreaseLM	45.1	72.4	94.9
DRAGON (**Ours**)	**47.5**	**73.4**	**96.4**

Table 4.7: Accuracy on biomedical NLP tasks. DRAGON outperforms all previous biomedical LMs.

Downstream evaluation tasks. We finetune and evaluate DRAGON on three popular biomedical NLP and reasoning benchmarks: MedQA-USMLE (**MedQA**) (Jin et al., 2021b), **PubMedQA** (Jin et al., 2019), and **BioASQ** (Nentidis et al., 2019). §4.8.2 provides details on these tasks and data splits.

Baselines. We compare DRAGON with the vanilla LM (BioLinkBERT) and LMs finetuned with the KG (QAGNN and GreaseLM seeded with BioLinkBERT).

Results. Table 4.7 summarizes model performance on the downstream tasks. Across tasks, DRAGON outperforms all the existing biomedical LMs and KG-augmented QA models, e.g., +3% absolute accuracy boost over BioLinkBERT and +2% over GreaseLM on *MedQA*, achieving new state-of-the-art performance on these tasks. This result suggests significant efficacy of KG-augmented pretraining for improving biomedical reasoning tasks. Combined with the results in the general commonsense domain (§4.4.4), our experiments also suggest the domain-generality of DRAGON, serving as an effective pretraining method across domains with different combinations of text, KGs and seed LMs.

4.6 Conclusion

We presented DRAGON, a self-supervised pretraining method to learn a deeply bidirectional language-knowledge model from text and knowledge graphs (KGs) at scale. In both general and biomedical domains, DRAGON outperforms existing language models and KG-augmented models on various NLP tasks, and exhibits strong performance on complex reasoning such as answering questions involving long context or multi-step reasoning.

4.7 Ethics

We outline potential ethical considerations. First, DRAGON is a method to fuse language representations and knowledge graph representations for joint reasoning. Consequently, DRAGON may reflect the same biases and toxic behaviors exhibited by language models and knowledge graphs that are used to initialize it. For example, it is shown that language models many encode biases about demographic attributes (Sheng et al., 2020; Weidinger et al., 2021) and generate toxic outputs (Gehman et al., 2020). Because DRAGON is seeded with pretrained language models, it is possible to reflect them in open-world settings. Second, the ConceptNet knowledge graph (Speer et al., 2017) used in this work has been shown to encode stereotypes (Mehrabi et al., 2021), rather than completely clean commonsense knowledge. If DRAGON were used outside these standard benchmarks in conjunction with ConceptNet as a KG, it might rely on unethical relationships in its knowledge resource to arrive at conclusions. Consequently, while DRAGON could be used for applications

outside these standard benchmarks, we would encourage implementers to use the same precautions they would apply to other language models and methods that use noisy knowledge sources.

Another ethical consideration is the use of the MedQA-USMLE evaluation. While we find this clinical reasoning task to be an interesting testbed for DRAGON and for joint language and knowledge reasoning in general, we do not encourage users to use these models for real world clinical prediction.

4.8 Supplementary

4.8.1 Experimental Setup Details

KG retrieval

Given each input text segment W, we follow the procedure from Yasunaga et al. (2021) to retrieve a relevant local KG G from the raw KG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. First, we use the entity linker from the spaCy library[1] to link entity mentions in W to entity nodes in $\mathcal{G}$, obtaining an initial set of nodes V_{el}. Second, we add any bridge entities in $\mathcal{G}$ that are in a 2-hop path between any pair of linked entities in V_{el} to get the total retrieved nodes $V \subseteq \mathcal{V}$. If the number of nodes in V exceeds 200, we prune V by randomly sampling 200 nodes from it to be the final retrieved nodes V. Lastly, we retrieve all the edges in $\mathcal{G}$ that connect any two nodes in V to obtain $E \subseteq \mathcal{E}$, forming the final local KG, $G = (V, E)$.

Graph initialization

For the ConceptNet knowledge graph used in the general commonsense domain (§4.4), we follow the method of MHGRN (Feng et al., 2020) to prepare the initial KG node embeddings. Specifically, we convert triplets in the KG into sentences using pre-defined templates for each relation. Then, these sentences are fed into BERT-Large (Devlin et al., 2019) to compute embeddings for each sentence. Finally, for each entity, we collect all sentences containing the entity, extract all token representations of the entity's mention spans in these sentences, and return the mean pooling of these representations.

For the UMLS knowledge graph used in the biomedical domain (§4.5), node embeddings are initialized similarly using the pooled token output embeddings of the entity name from BioLinkBERT (Yasunaga et al., 2022b).

While extremely rare (¡ 1%), in case when the input text does not yield any linked entity, we represent the graph using a dummy node initialized with 0, i.e., DRAGON backs off to only using the text side representations because the graph propagates no information.

Hyperparameters

[1] https://spacy.io/

Category	Hyperparameter	Commonsense domain		Biomedical domain	
		Pretrain	Finetune	Pretrain	Finetune
Model architecture	Number of text-KG fusion layers M	5	5	5	5
	Number of Unimodal LM layers N	19	19	19	19
	Number of attention heads in GNN	2	2	2	2
	Dimension of node embeddings and the messages in GNN	200	200	200	200
	Dimension of MLP hidden layers (except MInt operator)	200	200	200	200
	Number of hidden layers of MLPs	1	1	1	1
	Dimension of MInt operator hidden layer	400	400	400	400
Regularization	Dropout rate of the embedding layer, GNN layers and dense layers	0.2	0.2	0.2	0.2
Optimization	Learning rate of parameters in LM	2e-5	{1e-5, 2e-5, 3e-5}	2e-5	{1e-5, 2e-5, 3e-5}
	Learning rate of parameters not in LM	3e-4	{3e-4, 1e-3}	3e-4	{1e-4, 3e-4}
	Number of epochs in which LM's parameters are kept frozen	2	4	2	4
	Optimizer	RAdam	RAdam	RAdam	RAdam
	Learning rate schedule	linear warmup and decay	linear warmup and decay	linear warmup and decay	linear warmup and decay
	Warmup ratio	0.1	0.1	0.1	0.1
	Batch size	8,192	128	8,192	128
	Number of epochs	-	10–70	-	10–70
	Number of steps	20,000	-	20,000	-
	Max gradient norm (gradient clipping)	1.0	1.0	1.0	1.0
Data	Max number of nodes	200	200	200	200
	Max number of tokens	512	{128, 256}	512	512

Table 4.8: Hyperparameter settings for models and experiments

Table 4.8.

4.8.2 Downstream Evaluation Tasks

We use the following nine commonsense reasoning benchmarks for the experiments in the general domain (§4.4).

CommonsenseQA (CSQA) (Talmor et al., 2019) is a 5-way multiple-choice QA task testing commonsense reasoning. The dataset has 12,102 questions. We use the in-house data splits by (Lin et al., 2019).

OpenbookQA (OBQA) (Mihaylov et al., 2018) is a 4-way multiple-choice QA task containing elementary science questions. It has 5,957 questions. We use the original data splits in (Mihaylov and Frank, 2018).

RiddleSense (Riddle) (Lin et al., 2021) is a 5-way multiple-choice task testing complex riddle-style commonsense reasoning. It has 5,715 questions. We split the dev set in half to make in-house dev/test sets.

AI2 Reasoning Challenge, Challenge Set (ARC) (Clark et al., 2018) is a 4-way multiple-choice QA task containing science exam questions. It has 2,590 questions. We use the original data splits in (Clark et al., 2018).

CosmosQA (Huang et al., 2019b) is a 4-way multiple-choice QA task testing commonsense reasoning with long narratives. It has 35.6K questions. We split the dev set in half to make in-house dev/test sets.

HellaSwag (Zellers et al., 2019) is a 4-way multiple-choice task testing grounded commonsense reasoning about events. It has 70K questions. We split the dev set in half to make in-house dev/test sets.

Physical Interaction QA (PIQA) (Bisk et al., 2020) is a 3-way multiple-choice QA task testing physics reasoning about objects. It has 20K questions. We split the dev set in half to make in-house dev/test sets.

Social Interaction QA (SIQA) (Sap et al., 2019) is a 3-way multiple-choice QA task testing social commonsense reasoning. It has 37K questions. We use the original data splits in (Sap et al., 2019).

Abductive Natural Language Inference (aNLI) (Bhagavatula et al., 2020) is a 2-way multiple-choice task testing abductive commonsense reasoning. It has 170K questions. We use the original data splits in (Bhagavatula et al., 2020).

For the experiments in the biomedical domain (§4.5), we use the following three biomedical NLP and reasoning benchmarks.

MedQA-USMLE (MedQA) (Jin et al., 2021b) is a 4-way multiple-choice task containing United States Medical License Exam questions. The dataset has 12,723 questions. We use the original data splits in (Jin et al., 2021b).

PubMedQA (Jin et al., 2019) is a 3-way multiple-choice task testing biomedical language understanding and reasoning. The dataset has 1,000 questions. We use the original data splits in (Jin et al., 2019).

BioASQ (Nentidis et al., 2019) is a 2-way multiple-choice task testing biomedical language understanding and reasoning. The dataset has 885 questions. We use the original data splits in (Nentidis et al., 2019).

Dataset	Example
CommonsenseQA	A weasel has a thin body and short legs to easier burrow after prey in a what? (A) tree (B) mulberry bush (C) chicken coop (D) viking ship **(E) rabbit warren**
OpenbookQA	Which of these would let the most heat travel through? (A) a new pair of jeans **(B) a steel spoon in a cafeteria** (C) a cotton candy at a store (D) a calvin klein cotton hat
RiddleSense	What home entertainment equipment requires cable? (A) radio shack (B) substation (C) cabinet **(D) television** (E) desk
AI2 Reasoning Challenge	Which property of a mineral can be determined just by looking at it? **(A) luster** (B) mass (C) weight (D) hardness
CosmosQA	It's a very humbling experience when you need someone to dress you every morning, tie your shoes, and put your hair up. Every menial task takes an unprecedented amount of effort. It made me appreciate Dan even more. But anyway I shan't dwell on this (I'm not dying after all) and not let it detact from my lovely 5 days with my friends visiting from Jersey. What's a possible reason the writer needed someone to dress him every morning? (A) The writer doesn't like putting effort into these tasks. **(B) The writer has a physical disability.** (C) The writer is bad at doing his own hair. (D) None of the above choices.
HellaSwag	A woman is outside with a bucket and a dog. The dog is running around trying to avoid a bath. She (A) rinses the bucket off with soap and blow dries the dog's head. (B) uses a hose to keep it from getting soapy. **(C) gets the dog wet, then it runs away again.** (D) gets into the bath tub with the dog.
Physical Interaction QA	You need to break a window. Which object would you rather use? **(A) a metal stool** (B) a giant bear (C) a bottle of water
Social Interaction QA	In the school play, Robin played a hero in the struggle to the death with the angry villain. How would others feel as a result? (A) sorry for the villain **(B) hopeful that Robin will succeed** (C) like Robin should lose the fight
aNLI	Obs1: It was a gorgeous day outside. Obs2: She asked her neighbor for a jump-start. **Hyp1: Mary decided to drive to the beach, but her car would not start due to a dead battery.** Hyp2: It made a weird sound upon starting.
MedQA-USMLE	A 57-year-old man presents to his primary care physician with a 2-month history of right upper and lower extremity weakness. He noticed the weakness when he started falling far more frequently while running errands. Since then, he has had increasing difficulty with walking and lifting objects. His past medical history is significant only for well-controlled hypertension, but he says that some members of his family have had musculoskeletal problems. His right upper extremity shows forearm atrophy and depressed reflexes while his right lower extremity is hypertonic with a positive Babinski sign. Which of the following is most likely associated with the cause of this patients symptoms? (A) HLA-B8 haplotype (B) HLA-DR2 haplotype **(C) Mutation in SOD1** (D) Mutation in SMN1
PubMedQA	Recent studies have demonstrated that statins have pleiotropic effects, including anti-inflammatory effects and atrial fibrillation (AF) preventive effects [...] 221 patients underwent CABG in our hospital from 2004 to 2007. 14 patients with preoperative AF and 4 patients with concomitant valve surgery [...] The overall incidence of postoperative AF was 26%. Postoperative AF was significantly lower in the Statin group compared with the Non-statin group (16%versus 33%, p=0.005). Multivariate analysis demonstrated that independent predictors of AF [...] Do preoperative statins reduce atrial fibrillation after coronary artery bypass grafting? **(A) yes** (B) no (C) maybe
BioASQ	LT4 absorption is unchanged by concomitant metformin ingestion. It has been hypothesized that metformin may suppress serum thyrotropin (TSH) concentrations by enhancing LT4 absorption or by directly affecting the hypothalamic-pituitary axis. Does metformin interfere thyroxine absorption? (A) yes **(B) no**

Table 4.9: Example for each downstream task dataset used in this work.

4.8.3 Additional Experimental Results

Method	**Hit@3**
DistMult (i.e., KG only)	61.3
DRAGON (i.e., KG + text)	**78.1**

Table 4.10: **KG link prediction performance on ConceptNet**. In addition to the NLP tasks we mainly used for downstream evaluation, DRAGON can also perform KG link prediction tasks in downstream. We find that DRAGON (which uses retrieved text besides the KG) achieves improved performance on the KG link prediction task compared to the baseline DistMult model (which does not use text).

Chapter 5

Fusing Visual Knowledge

In the previous chapters, we have presented methods to effectively fuse textual knowledge and structured knowledge bases into language models. In this chapter, we develop more unified models that also incorporate visual knowledge to enable various applications involving text and images.

5.1 Introduction

Recent multimodal models have achieved remarkable progress in image and text generation. DALL-E (Ramesh et al., 2021) and Parti (Yu et al., 2022b) perform image generation from text, Flamingo (Alayrac et al., 2022b) performs text generation from images, and CM3 (Aghajanyan et al., 2022) offers a unified Transformer model that generates both text and images. Typically, these models store all their knowledge (e.g., the appearance of the Eiffel Tower) implicitly in the parameters of the underlying neural network, requiring a lot of parameters (e.g., 10–80B) and training data (e.g., 1–10B images) to cover all the knowledge. This motivates the development of multimodal models that can refer to an external memory of knowledge (e.g., web data) for increased knowledge capacity. Access to external memory is useful to accommodate the growth and update of knowledge through time, and is especially helpful for tasks that involve entity knowledge, such as generating images for entity-rich captions like "George Washington standing in front of the Eiffel Tower". Reference to external memory may also offer benefits such as explainable and faithful predictions (Metzler et al., 2021).

Recently, retrieval-augmented language models have shown promise in natural language processing (NLP) (Karpukhin et al., 2020; Guu et al., 2020; Lewis et al., 2020c; Borgeaud et al., 2022). Given input text, such a model uses a *retriever* that retrieves relevant documents from an external memory, and uses a *generator* to generate predictions given the retrieved documents. However, these retrieval-augmented methods are studied originally for text, and extending them to the multimodal setting remains an open problem with challenges. Specifically, we need to design a retriever and a

generator that handle multimodal documents, consisting of *both* images and text. Several concurrent works study retrieval for multimodal data (Chen et al., 2022a;b), but their generators are each limited to a single modality, either text generation or image generation (Table 5.1).

Here, we address the above challenge and present the first retrieval-augmented multimodal model that can retrieve and generate *both* text and images. As in Figure 5.1, our input data and external memory comprise a set of *multimodal documents*, each of which is an arbitrary sequence of text/images (e.g., text, image, or their combinations like caption-image pair). First, to obtain a multimodal retriever, we use the Dense Retrieval method (Karpukhin et al., 2020) with a mixed-modal encoder that can encode combinations of text and images (e.g., pretrained CLIP; Radford et al. 2021b). Given this retriever, we design a technique to retrieve diverse and informative documents for the input document. Second, we design the retrieval-augmented generator based on the CM3 architecture (Aghajanyan et al., 2022), which is a Transformer sequence model capable of both text and image generation. Concretely, we prepend the retrieved documents as in-context examples to the main input document, and train the generator by optimizing token prediction loss jointly for the main document and retrieved documents.

We train our retrieval-augmented CM3 (*RA-CM3*), using 150M text-image pairs from the LAION dataset (Schuhmann et al., 2021). RA-CM3 achieves strong performance on MS-COCO image and caption generation, significantly outperforming the baseline CM3 with no retrieval (12 FID and 17 CIDEr improvements). It also outperforms existing models such as DALL-E and Flamingo, despite using fewer parameters ($<30\%$) and compute for training ($<30\%$).

We further demonstrate novel capabilities of RA-CM3 (§5.5). First, it can perform faithful generation for tasks that require entity knowledge, for which existing models struggle (Figure 5.3, 5.4). Second, RA-CM3 exhibits a multimodal in-context learning ability: it can perform controlled image generation by prompting with demonstration examples in context (Figure 5.7), and it can also perform few-shot image classification. RA-CM3 is the first model that can perform in-context learning for both text and image generation (Table 5.1).

More broadly, our work offers a general and modular retrieval augmentation framework for multimodal models, and opens up various research avenues, such as further advancement of multimodal retrievers and generators.

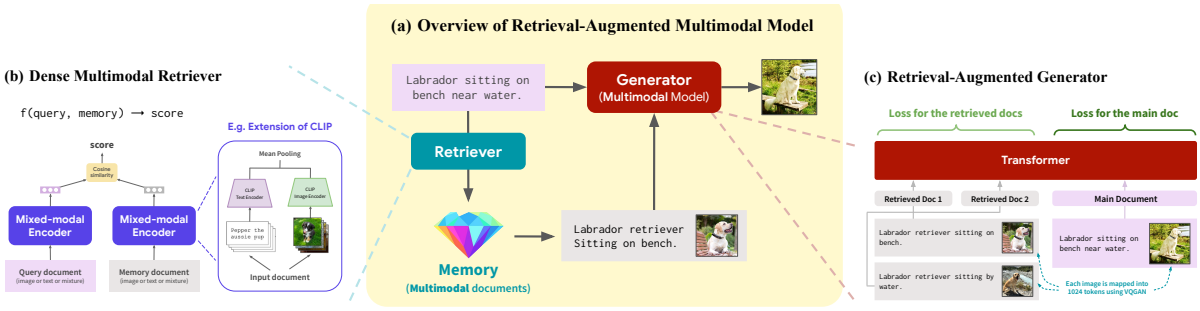


Figure 5.1: **Our approach, retrieval-augmented multimodal modeling**. **(a)** Overview: given an input multimodal document, we use a **retriever** to retrieve relevant multimodal documents from an external memory, and use the **generator** to refer to the retrieved documents and make predictions for the input (e.g., generate the continuation). **(b)** The multimodal retriever is a dense retriever with a mixed-modal encoder that can encode mixture of text and images (§5.3.2). **(c)** The retrieval-augmented generator uses the CM3 Transformer architecture, and we prepend the retrieved documents to the main input document that we feed into the model (§5.3.3).

Approach	Model type	Image generation	Text generation	Retrieval	In-context learning
DALL-E, Parti (Ramesh et al., Yu et al.)	Autoregressive	✔			
DALL-E 2, Imagen (Ramesh et al., Saharia et al.)	Diffusion	✔			
Re-Imagen (Chen et al.)	Diffusion	✔		✔	
Flamingo, MetaLM (Alayrac et al., Hao et al.)	Autoregressive		✔		✔
MuRAG (Chen et al.)	Autoregressive		✔†	✔	
CM3 (Aghajanyan et al.)	Autoregressive	✔	✔		
RA-CM3 (Ours)	Autoregressive	✔	✔	✔	✔

Table 5.1: **Comparison with other multimodal models**. Our RA-CM3 is the first retrieval-augmented model that can perform both image and text generation. RA-CM3 also exhibits strong in-context learning abilities thanks to the proposed retrieval-augmented training (§5.3.3). †Focus on question answering.

5.2 Related work

Vision-language multimodal models. Various models have been developed for text-to-image generation. Typically, these models are autoregressive Transformer-based, e.g., DALL-E (Ramesh et al., 2021) and Parti (Yu et al., 2022b), or diffusion-based, e.g., Imagen (Saharia et al., 2022), DALL-E 2 (Ramesh et al., 2022) and Stable Diffusion (Rombach et al., 2022). Meanwhile, several works also study image-to-text generation (Cho et al., 2020; Wang et al., 2022c). In particular, Flamingo (Alayrac et al., 2022b) is a Transformer-based image-to-text generation model, with in-context learning ability. Recently, CM3 (Aghajanyan et al., 2022) provides a unified model that uses a Transformer to perform both text and image generation. To make use of this generality, we will build on CM3 to design our model.

While the above models have achieved strong performance on image and text generation, they store all their knowledge inside the model, which tends to require a lot of parameters (e.g., 10B) and training data (e.g., 1B images). To address this limitation, we augment them with an ability to refer to relevant examples from an external memory when generating images/text. With this augmentation, our model outperforms the existing models by using less training data (150M images), compute and parameters ($<30\%$) (§5.4).

Retrieval-augmented language models. Retrieval augmentation has shown promise in NLP (Lewis et al., 2020c). To incorporate knowledge into a language model (LM), this line of work retrieves documents relevant to input text from an external memory, and lets the LM (generator) use the retrieved documents to make more informed predictions. The external memory used is typically a collection of text passages (Hashimoto et al., 2018; Khandelwal et al., 2019; Karpukhin et al., 2020; Guu et al., 2020; Lewis et al., 2020c;a; Yasunaga et al., 2022b; Borgeaud et al., 2022; Shi et al., 2023b) or a structured knowledge base (Zhang et al., 2019; Agarwal et al., 2021; Xie et al., 2022; Yasunaga et al., 2021; 2022a). Here we generalize the scope of the retrieval-augmented LM framework and consider *multimodal* documents for both our input data and external memory, which can be arbitrary sequences of text and images.

Retrieval in multimodal models. Besides the retrieval augmentation for language models, recent works also study retrieval for computer vision models (Ashual et al., 2022; Blattmann et al., 2022; Gur et al., 2021; Sarto et al., 2022; Li et al., 2022b; Ramos et al., 2023; Wang et al., 2022b). More recently, some concurrent works study retrieval in multimodal models: Re-Imagen (Chen et al., 2022b) performs diffusion-based caption-to-image generation using retrieved images; MuRAG (Chen et al., 2022a) performs natural language question answering using retrieved images. While these works focus on generating a single modality, either text or image, we develop a general and unified model that can retrieve, encode, and generate *both* images and text (Table 5.1). Moreover, our retrieval-augmented training allows the generator model to acquire novel in-context learning

ability such as controlled image generation (§5.5).

5.3 Approach

We present a retrieval-augmented multimodal model that can retrieve and generate both text and images. As illustrated in Figure 5.1, given an input multimodal document (i.e., arbitrary sequence of text/images), we use a **retriever** that retrieves relevant multimodal documents from an external memory, and uses a **generator** to refer to the retrieved documents and make predictions for the input document (i.e., generate the continuation). We design the multimodal retriever as a dense retriever with a mixed-modal encoder that can encode combinations of text and images (e.g., pretrained CLIP; §5.3.2). We build the retrieval-augmented generator using the CM3 Transformer architecture, and we prepend the retrieved documents to the main input document that we feed into the generator (§5.3.3). We describe how we train this model and use it for text-to-image or image-to-text generation in §5.3.4. Notably, our resulting model, *Retrieval-Augmented CM3 (RA-CM3)*, is the first multimodal model that can retrieve and generate *combinations* of text and images, which is the most general capability among existing multimodal models (Table 5.1). Moreover, while we build on existing techniques such as CLIP and CM3, we are the first to establish a method to unify them into a performant retrieval-augmented model through extensive analyses of design choices (§5.6.3).

5.3.1 Preliminaries

Retrieval augmented language model. The framework consists of a retrieval module R and a generator module G (e.g., language model). The retrieval module R takes an input sequence x and an external memory of documents $\mathcal{M}$, and returns a list of documents $M \subseteq \mathcal{M}$. The generator G then takes the input sequence x and the retrieved documents $M = (m_1, ..., m_K)$, and returns the target y, where y is the continuation of x in a typical language modeling task.

Causal masked multimodal model (CM3). CM3 (Aghajanyan et al., 2022) is a Transformer decoder (Vaswani et al., 2017) model for multimodal documents. A *multimodal document* is defined as an arbitrary sequence of text/images (e.g., text, image, or their combinations like caption-image pair). In particular, CM3 formats each multimodal document as an HTML sequence, such as "`<img alt=[text] src=[image]>`", where `[text]` is a sequence of text tokens, and `[image]` is a sequence of image tokens obtained by an image tokenizer such as VQGAN (Esser et al., 2021), which maps a raw image into 1024 tokens.

At training time, CM3 either takes the original sequence as input (e.g., x_{input} = "Photo of a cat: `[image]`") or converts it into an infilling instance by masking some spans and moving them to the end (e.g., x_{input} = "Photo of `<mask>`: `[image]` `<infill>` a cat"), and then optimizes the standard next token prediction loss for the input, $-\log p(x_{\text{input}})$. This provides a flexible model that

learns to perform infilling besides standard autoregressive generation. In particular, the model can perform both image and text generation: for caption-to-image, CM3 generates a continuation from the prompt "Photo of a cat:". For image-to-caption, CM3 generates from the prompt "Photo of `<mask>: [image] <infill>`".

Our setup. We aim to generalize the retrieval-augmented language model framework to the multimodal setting. Our input x + target y will be a multimodal document, and our memory $\mathcal{M}$ will be a set of multimodal documents. We design the retrieval module R for multimodal data (§5.3.2), and design the multimodal generator G based on CM3 (§5.3.3).

5.3.2 Multimodal retrieval

Dense retriever. A retriever r takes a query q (e.g., the input sequence x) and a candidate document m from the memory $\mathcal{M}$, and returns a relevance score $r(q, m)$. We follow the Dense Retrieval method (Karpukhin et al., 2020), in which the retriever r is a bi-encoder architecture,

$$r(q, m) = E_Q(q)^\top E_M(m) \tag{5.1}$$

where the query encoder E_Q and memory encoder E_M produce dense vectors for the query and memory document, respectively (Figure 5.1b). As our input and memory are multimodal documents, we let E_Q and E_M be **mixed-modal encoders** that encode a combination of text and images. While any mixed-modal encoders could be used in our framework, we find that a simple extension of CLIP (Ramesh et al., 2021) works well empirically, so we adopt it in our final system. Concretely, as shown in Figure 5.1b (right), given a multimodal document, we split it into a text part and an image part, encode the two parts separately using off-the-shelf frozen CLIP text and image encoders, and then average the two, with the L2 norm scaled to 1, as the vector representation of the document. We use this encoding method for both E_Q and E_M. Intrinsic evaluation of this CLIP-based retriever can be found in §5.6.1.

Given this retriever r, we perform Maximum Inner Product Search (MIPS; §5.4.1) over the memory to obtain a list of candidate documents sorted by the relevance score. We then sample the final K retrieved documents from this list.

Retrieval strategy. We discuss three key factors in obtaining/sampling informative retrieved documents for the generator in practice.

Relevance: The retrieved documents need to be relevant to the input sequence; otherwise, the retrieved documents do not provide useful information for modeling the main input sequence (see §5.6.3 for the ablation study). The dense retriever score based on CLIP captures this relevance factor.

Modality: While existing works on retrieval (Chen et al., 2022b) typically retrieve either an image or text only for the generator, we find that retrieving a multimodal document that consists of both images and text leads to better generator performance (see §5.6.3). Our intuition is that a multimodal document can be more informative because the text and image within it can contextualize each other. Hence, in our final system, we retrieve the raw multimodal documents that keep both images and text for the generator.

Diversity: We find that ensuring diversity in retrieved documents is important. First, simply sampling or taking the top K from the document list based on the relevance score can result in duplicate or highly similar images or text, leading to poor generator performance. This is especially important in the multimodal setting because even when two multimodal documents are not duplicates by themselves, the images or text contained in them can be duplicates, hurting the generator performance. To avoid redundancy, when we take documents from the top of the list, we skip a candidate if it is too similar (e.g., relevance score > 0.9) to the query or to the documents we already retrieved. Second, to further encourage diversity, we also propose Query Dropout, which drops some tokens of the query used in retrieval (e.g., 20% of tokens). This technique serves as regularization for training, and leads to further improvement in generator performance. Hence, our final system uses these two techniques (Avoid Redundancy + Query Dropout) for training, and uses Avoid Redundancy for inference. See §5.6.3 for detailed analysis.

5.3.3 Multimodal generator

We use CM3 as the base of our multimodal generator G. To incorporate the retrieved documents $M = (m_1, ..., m_K)$ into the generator, we prepend them to the main input sequence x, and feed the resulting sequence $(m_1, ..., m_K, x)$ to the Transformer (Figure 5.1c). In other words, the retrieved documents are in-context examples for the main input.

To train the generator, we optimize the following loss:

$$L = L_{\text{main}} + \alpha L_{\text{retr}} \tag{5.2}$$

$$= -\log p(x \mid m_1, ..., m_K) - \alpha \log p(m_1, ..., m_K) \tag{5.3}$$

where L_{main} and L_{retr} are the CM3 token prediction loss for the main input sequence x and for the retrieved documents $(m_1, ..., m_K)$, respectively. Here we propose optimizing the two loss terms jointly, with $\alpha \geq 0$. Existing retrieval-augmented models (e.g., Lewis et al. 2020c) typically only optimize the loss for the main sequence, L_{main} (i.e., $\alpha = 0$). However, as the Transformer computes logits for tokens in the retrieved documents when it computes logits for tokens in the main sequence, we can easily include the loss for the retrieved documents, L_{retr}. Thus, $\alpha > 0$ offers an effect analogous to increasing the batch size (the number of tokens involved in optimization) without much extra compute, and boosts training efficiency. This technique is especially useful in the multimodal

modeling setting, because each image takes many tokens (e.g., 1024 tokens), and $\alpha = 0$ would throw away computation used for the image tokens in retrieved documents. In practice, we find $\alpha = 0.1$ works well. See §5.6.3 for detailed analysis.

5.3.4 Training and inference

Training. Given a full input document x, we use either its text part or its image part as the query q for retrieving documents (§5.3.2). We then optimize the generator token prediction loss over the whole concatenated sequence (Equation 5.2) by standard teacher forcing. We only use the text or image part as the query because (1) retrieving documents based on the full input document could make the generator's token prediction task too easy during training, and (2) this training setting is close to the typical inference scenarios of text-to-image and image-to-text generation.

Since our off-the-shelf CLIP-based retriever already performs well, we fix the retriever and only train the generator in this work. An interesting future research direction would be the exploration of co-training or fine-tuning the retriever.

Inference. Our method takes an input sequence (prompt) x, uses x as the query for retrieval, and then lets the generator take the retrieved documents as part of the input to decode the continuation of x. For instance, for text-to-image generation, prompt x takes the source caption, and the continuation will be the target image. For image-to-text, prompt x takes the source image, and the continuation will be the target caption. Thus, the retriever only uses the prompt as a query and never sees the ground-truth continuation y to be evaluated, ensuring no information leakage.

5.4 Experiments

To experiment with our proposed approach, we train models using the LAION mutlimodal dataset (§5.4.1), and evaluate on the MS-COCO image and caption generation tasks (§5.4.2). We show that our retrieval-augmented model (RA-CM3) significantly improves both image and text generation performance (§5.4.3). We then analyze the scaling laws and key design choices of our model (§5.6.2, 5.6.3). Finally, §5.5 presents qualitative results and capabilities of our model, such as knowledge intensive generation and in-context learning.

5.4.1 Training setup

Data. To train our model, we use LAION (Schuhmann et al., 2021), an open-sourced dataset that consists of text-image pairs collected from the web. Following the preprocessing step of Stable Diffusion (Rombach et al., 2022), we cleaned a subset of LAION[1] and obtained 150M text-image pairs

[1] We filter out images with watermark probability above 0.5, unsafe probability above 0.5, or resolution below 256 $\times$ 256.

in total. Following CM3, we format each text-image pair as an HTML document, "`<img alt=[text] src=[image]>`", where `[image]` is a sequence of 1024 image tokens obtained by tokenizing the raw image using VQGAN (Esser et al., 2021; Gafni et al., 2022). These 150M documents are used as our model's final training data.

We also use the same 150M documents for our external memory $\mathcal{M}$.

Implementation. In our retrieval module R, we use the off-the-shelf CLIP model (`ViT-L/14`) (Radford et al., 2021b) for both the query and memory encoders E_Q and E_M. We use FAISS (Johnson et al., 2019) to index the external memory $\mathcal{M}$ (Flat Index) and perform MIPS-based retrieval.

For our generator G, we use a Transformer (Vaswani et al., 2017) of 2.7B parameters. The sequence length is 4096, which can take up to 3 documents. For each input document x, we retrieve $K \sim \text{Uniform}(\{0, 1, 2\})$ documents and prepend them to x. At inference time, we may also retrieve and add $K > 2$ documents via ensemble (see §5.5.4).

The model is trained from scratch for five days on 256 A100 GPUs. Our implementation is in PyTorch (Paszke et al., 2019) using Metaseq (Zhang et al., 2022a). We use model parallelism over 4 GPUs and a batch size of 16 sequences per GPU. The optimization uses a linear learning rate decay with 1500 warmup steps, a peak learning rate of 1e-4, a gradient clipping of 1.0, and the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ (Kingma and Ba, 2015).

Approach	Model type	MS-COCO FID (↓)	
		Not finetuned	Finetuned
Retrieval Baseline	-	17.97	-
KNN-Diffusion (Ashual et al., 2022)	Diffusion	16.66	-
Stable Diffusion (Rombach et al., 2022)	Diffusion	12.63	-
GLIDE (Nichol et al., 2021)	Diffusion	12.24	-
DALL-E 2 (Ramesh et al., 2022)	Diffusion	10.39	-
Imagen (Saharia et al., 2022)	Diffusion	7.27	-
Re-Imagen (Chen et al., 2022b)	Diffusion	6.88	5.25
DALL-E (12B) (Ramesh et al., 2021)	Autoregressive	~28	-
CogView (4B) (Ding et al., 2021)	Autoregressive	27.1	-
CogView2 (6B) (Ding et al., 2022)	Autoregressive	24.0	17.7
Parti (20B) (Yu et al., 2022b)	Autoregressive	7.23	3.22
Vanilla CM3	Autoregressive	29.5	-
RA-CM3 (2.7B) (Ours)	Autoregressive	**15.7**	-

Table 5.2: **Caption-to-image generation performance on MS-COCO**. Our retrieval-augmented CM3 significantly outperforms the baseline CM3 with no retrieval, as well as other models such as DALL-E (12B parameters). Moreover, our model achieves strong performance with much less training compute than existing models; see Figure 5.2 for details.

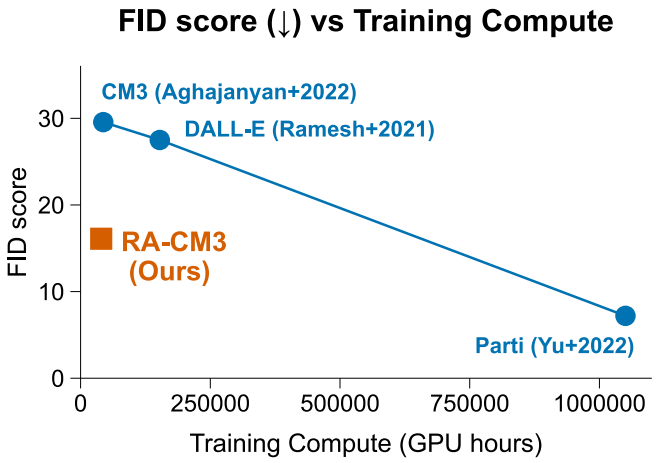


Figure 5.2: **Image generation quality vs training compute** for our RA-CM3 model and baseline models. x-axis is the amount of training compute used in terms of A100 GPU hours. y-axis is the MS-COCO FID score (the lower, the better). Our retrieval-augmented method achieves significantly better training efficiency than existing works under a similar autoregressive Transformer paradigm (e.g., CM3, DALL-E, Parti).

Baseline. For our baseline, we train a vanilla CM3 with no retrieval augmentation, using the same model architecture, training data, and amount of compute, for fair comparison. Since RA-CM3's external memory consists of the same training data, the total information accessible to RA-CM3 and vanilla CM3 are controlled to be the same.

5.4.2 Evaluation setup

For the main evaluation, we use the standard benchmark, MS-COCO (Lin et al., 2014), to evaluate both text-to-image and image-to-text generation. We evaluate our trained model with no further finetuning.

For text-to-image, following prior works (Ramesh et al., 2021; Nichol et al., 2021), we generate images for the MS-COCO validation set captions and measure the FID score (Heusel et al., 2017) against ground-truth images. To generate an image for each caption, we sample 10 images from the model and then take the top image based on the CLIP score (Radford et al., 2021b) with respect to the input caption, as done in Aghajanyan et al. (2022).

For image-to-text, following prior works (Alayrac et al., 2022b), we generate captions for the MS-COCO validation set images and measure the CIDEr score (Vedantam et al., 2015) against ground-truth captions. To generate an caption for each image, we sample 32 captions from the model and take the top caption based on perplexity (Fried et al., 2022).

Approach	**CIDEr (↑)**
Retrieval Baseline	84.1
DALL-E$^{\text{Small}}$ (Wang, 2021)	20.2
ruDALL-E-XL (Forever, 2021)	38.7
minDALL-E (Kim et al., 2021a)	48.0
X-LXMERT (Cho et al., 2020)	55.8
Parti (Yu et al., 2022b)	83.9
Flamingo (3B; 4-shot) (Alayrac et al., 2022b)	85
Flamingo (80B; 4-shot) (Alayrac et al., 2022b)	103
Vanilla CM3	71.9
RA-CM3 (2.7B) (Ours)	**89.1**

Table 5.3: **Image-to-caption generation performance on MS-COCO** (with no finetuning). Our retrieval-augmented CM3 significantly outperforms the baseline CM3 with no retrieval. Moreover, our model outperforms other strong models such as Parti (20B parameters) and Flamingo (3B; 4-shot), despite using just ~3B parameters and 2-shot in-context examples.

5.4.3 Main results

Caption-to-image generation. Table 5.2 shows the caption-to-image generation performance on MS-COCO. The metric is FID score, where lower is the better. Our retrieval-augmented CM3 achieves an FID score of 16 without finetuning, significantly outperforming the baseline CM3 with no retrieval (FID 29) and other models such as DALL-E (FID 28), which is 3x bigger than our model. This suggests that retrieval augmentation provides significant help in generating higher-quality images.

To also factor in training efficiency, Figure 5.2 visualizes the image generation performance (y-axis: FID score) vs the amount of compute used in model training (x-axis: normalized A100 GPU hours) for our RA-CM3 model and baseline models. We find that existing models in the autoregressive Transformer paradigm follow a negatively sloped line in this chart (the blue dots and line in Figure 5.2). RA-CM3 is located significantly below this line, i.e., obtaining a better FID with less training compute. This suggests that the proposed retrieval-augmented method achieves significantly better training efficiency than existing works.

Our intuition is that retrieval augmentation allows the model to focus on learning how to use the retrieved documents in the context rather than fitting all the documents into the parameters of the model, speeding up the training process.

Image-to-caption generation. Table 5.3 shows the image-to-caption generation performance on MS-COCO, with no finetuning. The metric is the CIDEr score, where the higher is the better. Our retrieval-augmented CM3 achieves a CIDEr score of 89, significantly outperforming the baseline CM3 with no retrieval (CIDEr 72). Moreover, RA-CM3 outperforms other strong models such as Parti (20B parameters) and Flamingo (3B; 4-shot), despite using just ~3B parameters and 2-shot in-context examples.

These results confirm that our model can perform both image and text generation well, offering the first unified retrieval-augmented multimodal model (Table 5.1).

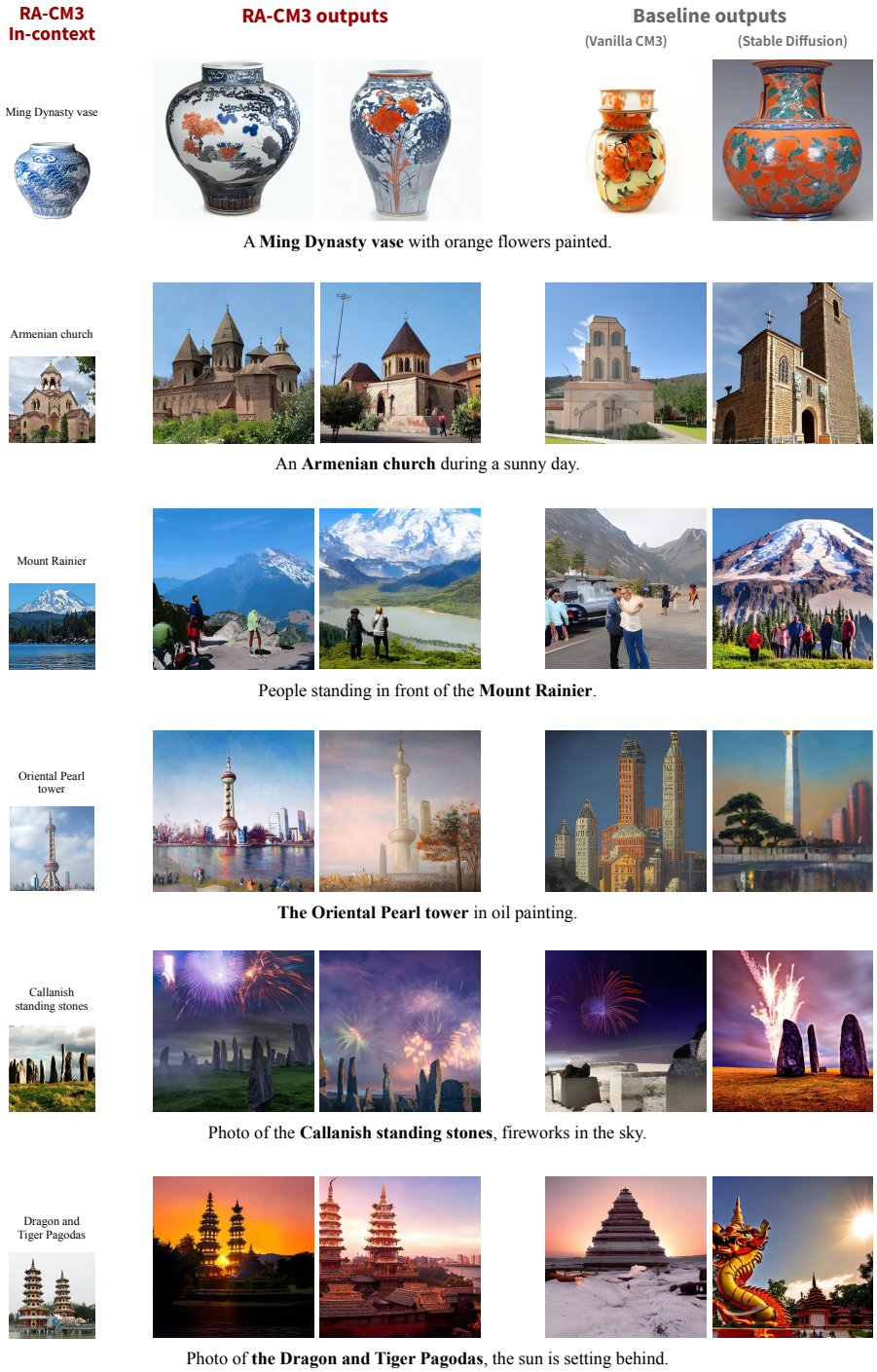


Figure 5.3: **Text-to-image generation involving world knowledge.** Our retrieval-augmented model (RA-CM3) can generate correct images from entity-rich captions thanks to the access to retrieved images in the context. For example, RA-CM3's outputs faithfully capture the visual characteristics of various entities (e.g., the shape and painting of Ming Dynasty vase, the amount of Callanish standing stones). On the other hand, baseline models without retrieval capabilities (vanilla CM3, Stable Diffusion) tend to struggle, especially when the caption involves rare entities (e.g., "Ming Dynasty vase", "Oriental Pearl tower", "Dragon and Tiger Pagodas").

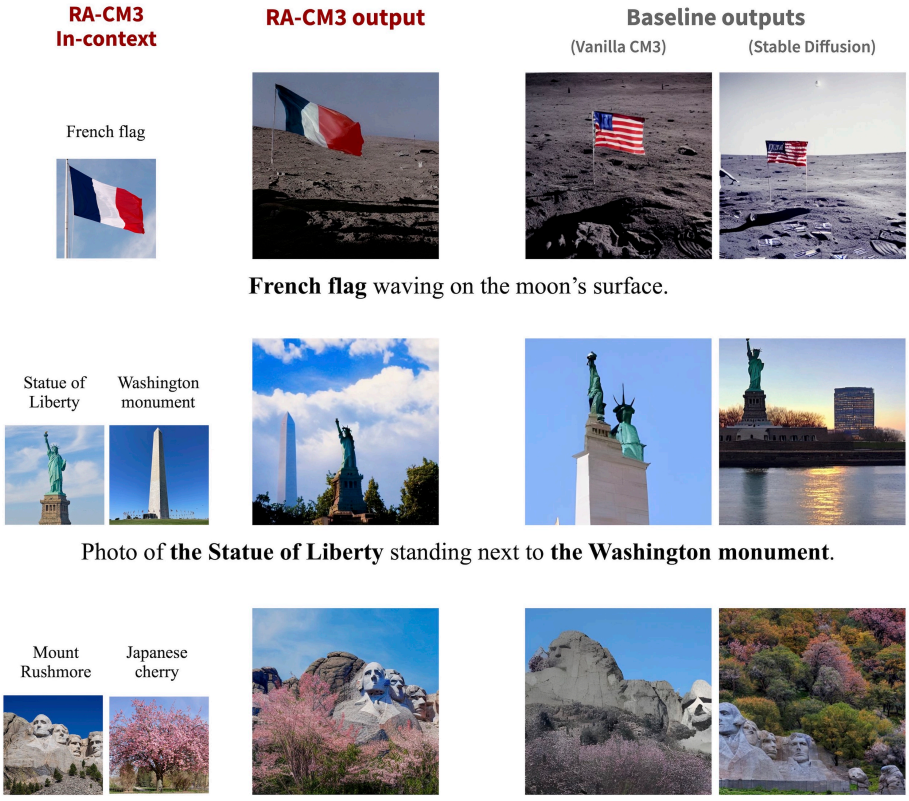


Figure 5.4: **Text-to-image generation involving rare *composition* of knowledge.** Our retrieval-augmented model (RA-CM3) can generate faithful images from captions that contain a rare or unseen composition of entities (e.g., "French flag" + "moon", "Mount Rushmore" + "Japanese cherry"). On the other hand, baseline models without retrieval capabilities (vanilla CM3, Stable Diffusion) tend to struggle on these examples, e.g., generate a US flag instead of a French flag on the moon.

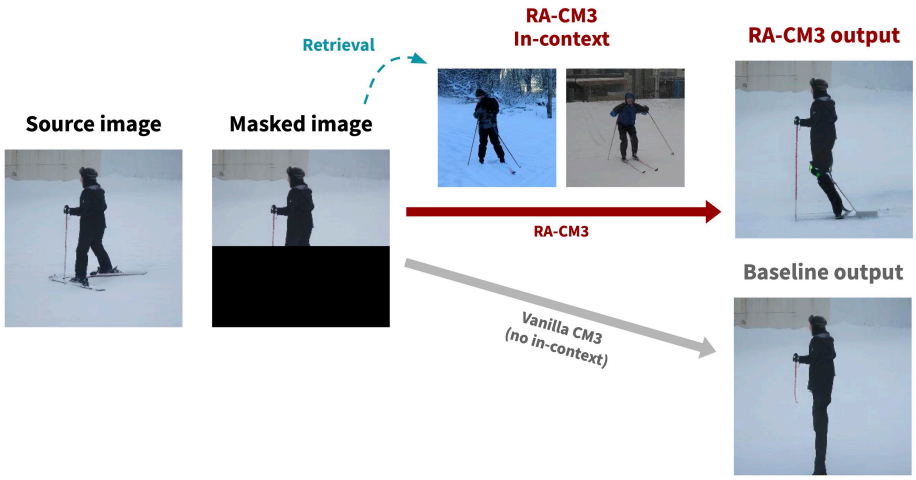


Figure 5.5: **Our model can perform better image infilling.** Infilling an image requires world knowledge, e.g., to recover the masked patches of the above image, the model needs to know about skiing. While the vanilla CM3 (no retrieval) tends to simply infill legs, our RA-CM3 (with retrieval) successfully recovers both legs and skis.

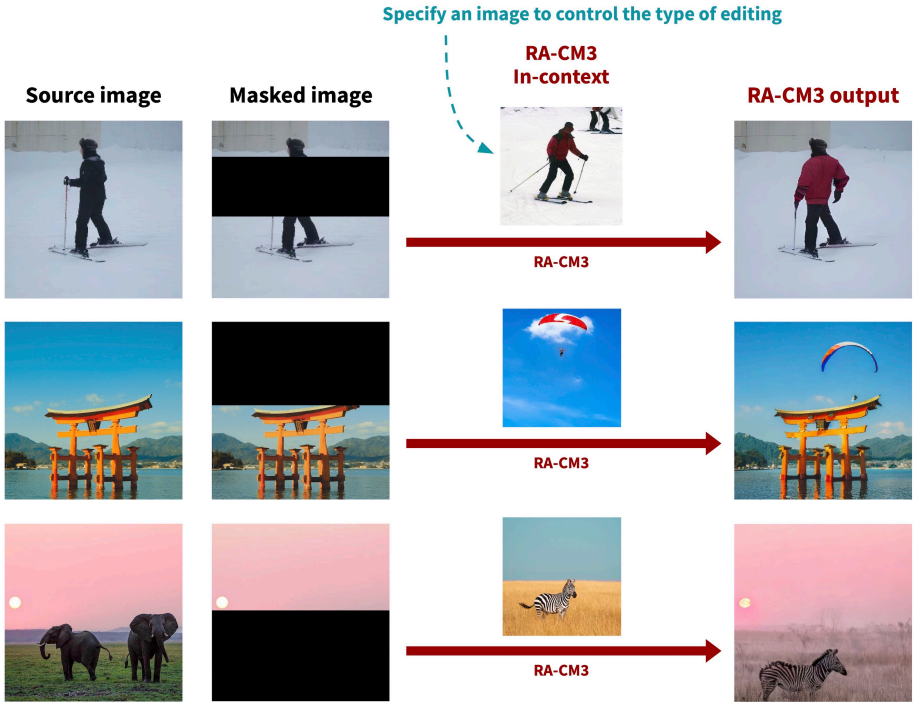


Figure 5.6: **Our model can perform image editing.** Instead of using retrieved examples in our RA-CM3's context (Figure 5.5), we can also intervene and manually specify the in-context examples to control image infilling. For instance, we can place an image of a person wearing a red jacket in the context to edit the black jacket in the original image to be red (Figure top).

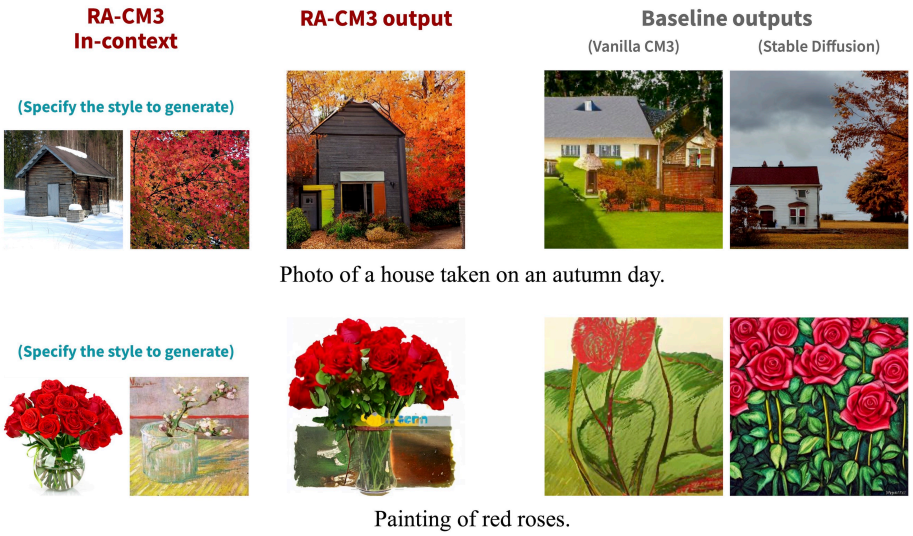
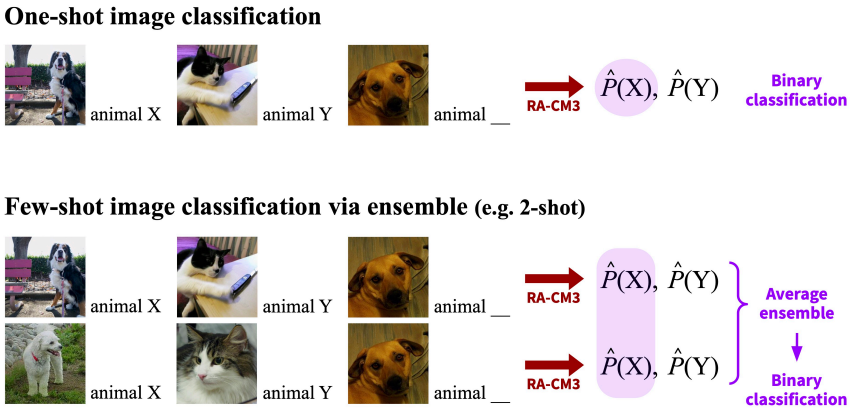


Figure 5.7: **Controllable image generation.** Our RA-CM3 model can control the style of caption-to-image generation by prepending demonstration examples in the generator's context. For instance, when generating an image of "a house taken on an autumn day" (Figure top), we can specify a concrete style by providing demonstration images (e.g., image of a triangular wooden house and image of orange autumn leaves background). Consequently, RA-CM3 generates an image that follows the visual characteristics of these in-context images.



Model	*k*-shot Accuracy			
	$k=1$	2	4	8
Baseline CM3	0.53	0.50	0.56	0.56
RA-CM3 (Ours)	**0.78**	**0.79**	**0.86**	**0.9**

Figure 5.8: **Our model performs one/few-shot image classification via in-context learning**. To assess the in-context learning ability, we consider a binary image classification task with non-semantic labels (e.g., "animal X" and "animal Y" instead of "dog" and "cat"). For one-shot classification (Figure top), we feed into the model one pair of demonstration examples, followed by a test example (`[test image]`, "animal _"), for which we predict the probability of "X" and "Y". For k-shot classification (Figure middle), we repeat the above procedure k times, each using a different pair of demonstration examples, and take the average ensemble of the predicted probability ("X" and "Y") across the k passes.

The table (Figure bottom) shows the results of k-shot classification accuracy, with $k = 1, 2, 4, 8$. Across all k's, our RA-CM3 improves on the baseline CM3 by large margins. Increasing k consistently improves accuracy for the k values above.

5.5 Qualitative results

We show novel qualitative capabilities of our RA-CM3, such as knowledge-intensive multimodal generation (§5.5.1) and multimodal in-context learning (§5.5.2, 5.5.3, 5.5.4). While GPT-3 (Brown et al., 2020) and Flamingo (Alayrac et al., 2022b) showed in-context learning for text-to-text or image-to-text generation, we show that RA-CM3 can do in-context learning for both text (§5.5.4) and image (§5.5.2, 5.5.3) generation.

5.5.1 Knowledge-intensive multimodal generation

Because of the retrieval capability, RA-CM3 is especially good at tasks that require world knowledge or composition of knowledge (knowledge-intensive generation). Figure 5.3, 5.4 show example outputs from RA-CM3. For each caption, the output images were obtained by sampling 256 images from the model and then re-ranking them using the CLIP score with respect to the input caption. We then apply an off-the-shelf super-resolution tool (Rombach et al., 2022).

World knowledge. Figure 5.3 shows model outputs for caption-to-image generation that involves world knowledge (e.g., specific entities). We find that our RA-CM3 model can generate correct images from entity-rich captions thanks to the access to retrieved images in the context. For example, RA-CM3's outputs faithfully capture the visual characteristics of various entities (e.g., the shape and painting of Ming Dynasty vase, the amount of Callanish standing stones). On the other hand, baseline models without retrieval capabilities (vanilla CM3, Stable Diffusion) tend to struggle, especially when the caption involves rare entities (e.g., "Ming Dynasty vase", "Oriental Pearl tower", "Dragon and Tiger Pagodas").

Composition of knowledge. Figure 5.4 shows model outputs for caption-to-image generation that involves rare *composition* of knowledge. We find that our retrieval-augmented model can generate faithful images from captions that contain a rare or unseen composition of entities (e.g., "French flag" + "moon", "Mount Rushmore" + "Japanese cherry"). On the other hand, baseline models without retrieval capabilities (vanilla CM3, Stable Diffusion) tend to struggle on these examples, e.g., generate a US flag instead of a French flag on the moon (Figure 5.4 top). This is likely because the US flag was the most common flag that co-occurred with the moon in the training data.

5.5.2 Image infilling and editing

Because our model builds on CM3, it can also perform *infilling*.[2] Figure 5.5 shows that our RA-CM3 can perform improved image infilling because of the retrieval capability. Infilling an image requires world knowledge, e.g., to recover the masked patches of the image in Figure 5.5, the model needs to

[2]To perform image infilling, the model is given "`[unmasked part of image] <mask> [unmasked part of image]`" as the prompt and generates the `<mask>` part as the completion. The final output image is constructed by plugging the generated output to the `<mask>` part of the input.

know about skiing. While the vanilla CM3 (no retrieval) tends to simply infill legs, RA-CM3 (with retrieval) successfully recovers both legs and skis.

Moreover, instead of using retrieved examples in the RA-CM3 context, we can also intervene and manually specify the in-context examples to control image infilling. Figure 5.6 shows examples. For instance, we can place an image of a person wearing a red jacket in the context to edit the black jacket in the original image to be red (Figure 5.6 top).

5.5.3 Controlled image generation

Controlled generation—controlling the behavior of models in generation (e.g., style of outputs)—is a key problem in generative models (Keskar et al., 2019; Li et al., 2019a).

Our RA-CM3 can control the style of caption-to-image generation by prepending demonstration examples in the generator's context (Figure 5.7). For instance, when generating an image for "Photo of a house taken on an autumn day" (Figure 5.7 top), we can specify a concrete style by providing demonstration images (e.g., an image of a triangular wooden house and an image of orange autumn leaves background). Consequently, RA-CM3 can generate an image that actually follows the visual characteristics of these in-context images. This is a very useful capability because we can control generation not only via text (captions) but also via image demonstrations—especially helpful when some visual characteristics we want to specify might be difficult to express in text.

Moreover, the finding that RA-CM3 can use in-context examples for controlled generation suggests that it has acquired a form of **multimodal in-context learning** ability. Our intuition is that because the RA-CM3 generator has seen relevant multimodal documents prepended to the main document in context during retrieval-augmented training, it has learned how to use in-context examples effectively.

5.5.4 One-shot and few-shot image classification

So far we have seen RA-CM3's in-context learning behavior for image generation (§5.5.3). Here we study its in-context learning ability for image-to-text generation, through one-shot and few-shot image classification.

Figure 5.8 illustrates the experiment. To assess the true in-context learning ability that factors out prior knowledge of the model, we consider a binary image classification task with non-semantic labels (e.g., "animal X" and "animal Y" instead of "dog" and "cat"). Specifically, we use ImageNet (Deng et al., 2009) to construct such evaluation sets where each class (e.g., animal X or Y) contains the same number of test images (e.g., 100 images). For one-shot classification (Figure 5.8 top), we feed into the model one pair of demonstration examples (`[image X]`, "animal X", `[image Y]`, "animal Y"), followed by a test example (`[test image]`, "animal _"), for which we predict the probability of "X" and "Y". For k-shot classification (Figure 5.8 middle), we repeat the above procedure k

times, each using a different pair of demonstration examples, and take the average ensemble of the predicted probability ("X" and "Y") across the k passes.[3]

The table in Figure 5.8 bottom shows the results of k-shot (binary) classification accuracy, with $k = 1, 2, 4, 8$. Across all k's, our RA-CM3 obtains significantly improved accuracy over the baseline CM3, which were not trained with retrieved documents in context. In particular, RA-CM3 already performs reasonably well at one-shot (0.78 accuracy at $k = 1$). This result suggests that RA-CM3 has acquired a strong in-context learning ability, especially given that we use non-semantic labels for image classes in this evaluation. Moreover, we find that increasing k consistently improves accuracy for the k values above (0.90 accuracy at $k = 8$). This observation suggests that ensemble is an effective method to increase the number of in-context examples to provide for the model.

5.6 Analysis

5.6.1 Intrinsic evaluation of CLIP-based retriever

Method	**Recall (↑)**		
	@1	@3	@5
CLIP text-to-image retrieval	48	65	78
CLIP text-to-mixture retrieval	61	77	85
CLIP image-to-text retrieval	56	75	84
CLIP image-to-mixture retrieval	78	84	87

Table 5.4: **Multimodal retrieval performance on MS-COCO**. We use the frozen pretrained CLIP. "text-to-image retrieval" and "image-to-text retrieval" use the CLIP text/image encoder as it is. "text-to-mixture retrieval" and "image-to-mixture retrieval" use our mixed-modal encoder based on CLIP (§5.3.2). In all these cases, the CLIP-based retrieval method performs reasonably well.

5.6.2 Scaling laws of RA-CM3

To study the scaling laws of retrieval augmentation for multimodal models, we train the retrieval-augmented CM3 and vanilla CM3 of various sizes (125M, 350M, 1.3B, 2.7B parameters) using the same amount of compute (two days on 256 GPUs), and then evaluate the models' perplexity on the MS-COCO validation set (Figure 5.9). We observe that RA-CM3 provides consistent improvements over vanilla CM3 across different scales. We do not observe any diminishing returns in the 125M–2.7B range that we studied. This suggests that retrieval augmentation is also promising at a larger

[3] An alternative way to use k-shot examples could be to prepend all the k pairs of demonstrations directly in RA-CM3's context, but this would take a significant sequence length in Transformer and might not be easy to scale. We find that the ensemble-based method performs well empirically, and comes with benefits such as being more scalable (parallel runs of shorter-length passes) and principled (agnostic to the order of the k examples).

scale.

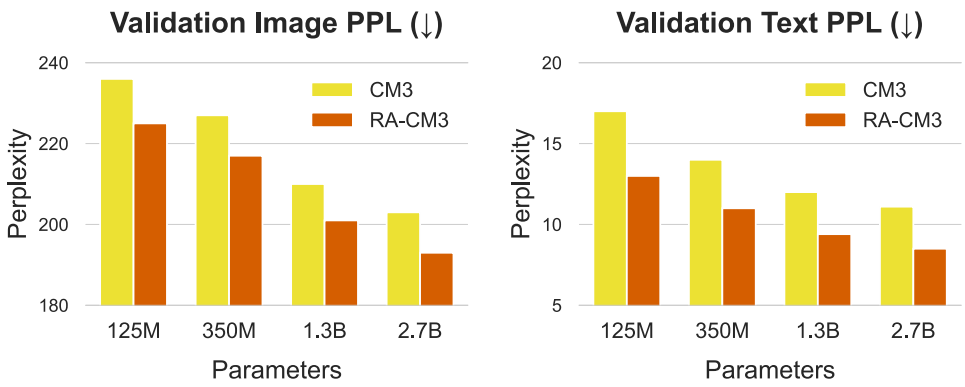


Figure 5.9: **Perplexity-based scaling laws for our RA-CM3 model.** We train RA-CM3 and vanilla CM3 of various parameter counts using the same amount of compute, and evaluate perplexity on the held-out validation set of MS-COCO. RA-CM3 provides consistent improvements over vanilla CM3 across different scales.

Method design	Choice	Image ppl (↓)	Text ppl (↓)
Retrieval relevance (§5.3.2)	Random at train & infer	246	23
	Retrieve at train, random at infer	246	24
	Random at train, retrieve at infer	243	18
	Retrieve at train & infer (**final**)	**227**	**13**
Retrieval modality (§5.3.2)	Only image or only text	234	15
	Multimodal document (**final**)	**227**	**13**
Retrieval diversity (§5.3.2)	Simply take top K	244	17
	Avoid redundancy	235	15
	Avoid redundancy + Query dropout (**final**)	**227**	**13**
Generator training (Equation 5.2)	$\alpha = 0$	239	17
	$\alpha = 1$	240	17
	$\alpha = 0.3$	231	14
	$\alpha = 0.1$ (**final**)	**227**	**13**

Table 5.5: **Analysis of our method's design choices.** As the metric, we use the perplexity of image/text generation on the MS-COCO validation set. We find that key methods to achieve the best performance are: ensure *relevance* in retrieved documents (table top); retrieve *multimodal* documents instead of only images or text (table second from top); encourage *diversity* in retrieved documents during training (table second from bottom); and train the token prediction loss for *both* the main input document and the retrieved documents, in particular, with a weight of $\alpha = 0.1$ (table bottom).
Note that images naturally have higher perplexity than text, as also observed in prior works (e.g., Aghajanyan et al. 2022).

5.6.3 Analysis of RA-CM3 designs

We analyze key design choices of the retrieval-augmented multimodal model, such as the strategies of retrieval (§5.3.2) and generator training (§5.3.3). Here, our experiments used a 2.7B parameter-model, trained for a day on 256 GPUs.

Retrieval relevance (Table 5.5 top). A main contribution of our work is retrieval-augmented training of the generator (§5.3.3). While our final RA-CM3 prepends documents retrieved by our CLIP-based retriever at both train and inference time ("Retrieve at train & infer" row in the table), one natural baseline is to train the model using random documents without retrieval (i.e., vanilla CM3) but use retrieved documents at inference time ("Random at train, retrieve at infer"). This baseline leads to a significant performance drop, suggesting that having relevant documents in context is crucial for model training. We also study other baselines, such as using retrieved documents at train time but random documents at inference time, or using random documents at both train and inference times. Both result in significant performance drops. These results confirm that relevance is a crucial factor in retrieval at both train and inference times.

Retrieval modality (Table 5.5 second from top). While existing works on retrieval (Chen et al., 2022b) typically retrieve either an image or text only for the generator, our retriever based on a mixed-modal encoder (§5.3.2), can retrieve a multimodal document that consists of both images and text. We find that retrieving multimodal documents performs better than retrieving only images or text. Our intuition is that a multimodal document can be more informative because the text and image within it can contextualize each other.

Retrieval diversity (Table 5.5 second from bottom). As discussed in §5.3.2, encouraging diversity in retrieved documents is important. Simply taking the top K (e.g., 2) from the list of candidate documents sorted by the retriever scores leads to poor performance—in fact, slightly worse than the baseline with no retrieval augmentation. Our first technique which avoids redundancy in retrieved documents leads to significant performance improvement. We also find that the second technique, Query Dropout, which encourages more diversity in retrieval during training leads to a further boost in evaluation performance.

Generator training (Table 5.5 bottom). A key design of our generator is that we optimize token prediction loss jointly for the main input document and the retrieved documents, with a weighting α (§5.3.3; Equation 5.2). Existing retrieval-augmented models typically optimize loss for the main document only ($\alpha = 0$), but we find that joint optimization ($\alpha > 0$) facilitates training and improves performance. We find that $\alpha = 0.1$ works well. Setting α to be too large (e.g., $\alpha = 1$) hurts training because this would place too much weight on modeling retrieved documents instead of the main document.

5.7 Discussions

5.7.1 Fair comparison of the retrieval-augmented model and non-retrieval-augmented model

Experiments: Our baseline is the vanilla CM3 with no retrieval augmentation. To make a fair comparison between the retrieval-augmented model (RA-CM3) and the baseline, RA-CM3 is trained using the same generator architecture, same training data, and same amount of compute as the vanilla CM3 (§5.4.1). RA-CM3's external memory used for retrieval also consists of the same training data. Thus, we ensured that no additional data or training compute is used for the retrieval-augmented model compared to the non-retrieval-augmented models. Under this controlled experiment, RA-CM3 substantially outperforms the vanilla CM3 in image and text generation (Table 5.2, 5.3). Figure 5.9 further indicates that RA-CM3 with fewer parameters (1.3B) can also outperform the vanilla CM3 with more parameters (2.7B). We also include the "Retrieval Baseline" in Table 5.2 and 5.3, which simply returns the retrieved images or text as model outputs. RA-CM3 outperforms this retrieval baseline.

Usage: Both the retrieval-augmented model and non-retrieval-augmented models take the same input from users, e.g., a source caption for image generation or a source image for caption generation. In the retrieval-augmented model, the retriever will automatically take this input prompt, fetch relevant images/text, and add them to the context of the generator, so no additional input is needed from the user. Of course, the user may also intervene and self-specify in-context examples for the generator, so the retrieval-augmented model provides more flexibility and controllability for users (§5.5.3). The retrieval step can be performed efficiently using FAISS (§5.4.1) in less than a second.

Thus, while the retrieval-augmented model operates within the same (fair) task definition as non-retrieval-augmented models, it opens up various benefits and new capabilities, such as improved explainability, faithfulness, and controllability in generation (§5.9).

5.7.2 Taking an existing model (e.g. vanilla CM3) and finetune it with retrieval-augmentation, instead of training the retrieval-augmented model (RA-CM3) from scratch

In practice, finetuning from the vanilla model can save compute for training RA-CM3 and is useful. The reason we trained RA-CM3 and vanilla CM3 both from scratch in our main experiments was to make a fair comparison between them by training with the same amount of compute, and to systematically study the effect of retrieval augmentation.

5.7.3 How the number of retrieved documents used for the generator (K) was set

We set K to be up to 2 (§5.4.1), primarily in consideration of the Transformer sequence length. Using the recent image tokenizer (e.g., Esser et al. 2021), each image is mapped to 1K tokens. Hence, the concatenation of $K=2$ retrieved documents and the main input document takes 3–4K tokens in total in Transformer. Increasing the Transformer sequence length beyond 4K incurs a significant burden in computation and GPU memory, so we decided on $K=2$. $K=2$ also worked reasonably well in practice (§5.4.3).

During inference, we may do ensemble (see §5.5.4) and take more than two retrieved documents in total into account. We conducted an analysis of varied $K=1,2,4,8$ in the MS-COCO caption-to-image generation evaluation (Table 5.6). We find that $K=2$ worked the best in this experiment. Our intuition is that most of the MS-COCO captions involve 1–2 objects, so retrieving the top two multimodal documents may be sufficient for generating corresponding images. It is an interesting future research to investigate larger K's on image generation tasks that involve more entities/objects.

Model	**Image Perplexity (↓)**			
	$K=1$	2	4	8
RA-CM3	228	227	228	232

Table 5.6: **MS-COCO caption-to-image generation performance** when the number of retrieved multimodal documents (K) is varied.

5.8 Conclusion

We presented a retrieval-augmented multimodal model that can retrieve and refer to an external memory for generating images and text. Specifically, we implemented a multimodal retriever using the pretrained CLIP and designed a retrieval-augmented generator using the CM3 architecture. Our resulting model, named RA-CM3, outperforms existing multimodal models on both image and caption generation tasks, while requiring much less training compute. Moreover, RA-CM3 exhibits novel capabilities such as knowledge-intensive image generation and multimodal in-context learning.

This work aims to offer a general and modular retrieval augmentation framework for multimodal models. We believe this opens up various exciting research avenues, such as improving the multimodal retriever and generator, extending modalities beyond image and text, and further investigating multimodal prompting and in-context learning.

5.9 Ethics and societal impact

Multimodal generative models, including our RA-CM3 and other models like DALL-E, Parti and Stable Diffusion, are typically trained on large, noisy image-text datasets collected from the web. These datasets may contain biases about demographic attributes, and unsafe content such as violence (Birhane et al., 2021a). We performed extensive data filtering to remove problematic content in training data following existing works (§5.4.1), but it may still be possible that RA-CM3 outputs problematic text or images. RA-CM3 is a **research prototype**, and we do not encourage using it in high-risk or sensitive domains or for generating images of people. For a more general, detailed discussion on the ethical considerations of multimodal generative models, we refer readers to e.g., Birhane et al. (2021b).

We also highlight the potential societal benefits of our retrieval-augmented multimodal model. First, as our model requires much less compute for training than existing models (§5.4.3), it can provide **energy savings**. Second, retrieval helps capture long-tail knowledge (e.g., rare entities or minority groups), which can contribute to more **fair** multimodal models. Third, retrieval naturally provides the provenance of knowledge, offering better interpretability and **explainability** about model predictions. Retrieval augmentation also helps make image/text generation **faithful** to the retrieved evidence documents (§5.5.1), potentially helping reduce unintentionally fake or hallucinated outputs.

Part II

Applications

In the previous part, we have developed methods to integrate various types of knowledge—textual, structured, and visual—in language models. In this part, we present the practical applications of our methods in clinical scenarios, such as clinical trial outcome prediction (Chapter 6) and multimodal medical question answering systems (Chapter 7).

Chapter 6

Clinical trials

In this chapter, we present the application and efficacy of textual and structured knowledge fusion (§3 and 4) in the challenging clinical task of clinical trial outcome prediction.

6.1 Introduction

A variety of different factors — environmental and biological at the molecular and cellular level — shape the treatment response. Same treatment may result in very different effectiveness and the likelihood of causing side effects when applied to different populations (Ramamoorthy et al., 2015; Schork, 2015; Charles et al., 2003; Siegel et al., 2000; Liu and Dipietro Mager, 2016; Franconi et al., 2007). For example, the bias towards testing drugs on younger male Caucasian participants has led to missed patient-safety markers, raising awareness about the importance of population properties in investigating treatment efficacy and safety (Knepper and McLeod, 2018). An overarching question is whether we can design more safe and effective treatments by changing the population properties to which the intervention is applied to (Liu et al., 2021b).

Current approaches for predicting population response to treatment typically focus on a a single disease and are designed for a specific task of interest (Jin et al., 2021a; Xu et al., 2019; Mobadersany et al., 2018). On the other hand, general approaches for predicting outcome of a treatment that capture large space of underlying biological interactions, typically as networks (Ruiz et al., 2021; Cheng et al., 2018; Luo et al., 2017; Cheng et al., 2019), do not account for variability between patients. As such, these approaches fail to model population or individual response to a particular treatment and are unable to discover interventions effective only in certain groups. Finally, existing approaches are unable to reason about factors that cause certain side effects or effectiveness of interventions (Santos et al., 2022). These approaches are typically black-box models that do not provide insights about causal relationships between interventions, population properties and the outcome.

Here, we present PlaNet, an extension of the DRAGON model (Yasunaga et al., 2022a) that predicts outcome of a treatment by reasoning over population variability, disease chemistry and drug biology. PlaNet is trained from a large clinical text corpus and a clinical knowledge graph that captures treatment information in form of the *(drug, condition, population)* triplets grounded in biomedical knowledge that captures underlying chemical and biological interactions. PlaNet first learns general-purpose representations of all treatment, biological and clinical entities in the knowledge graph in an unsupervised fashion. This is achieved by pretraining the model to capture the structure of the network and semantics of the terms. PlaNet can then be fine-tuned on many downstream pharmacological tasks.

We demonstrate the utility of PlaNet on clinical trials data. We structure the entire clinical trials database and incorporate it in PlaNet, resulting in a knowledge graph of $330,915$ nodes and $13,928,443$ heterogenous edges where population variability is described by clinical trials' eligibility criteria. We use PlaNet to predict outcome of clinical trials including trial efficacy as survival endpoint, likelihood of causing side effects, and exact side effect category. By representing knowledge as a graph, PlaNet is equally applicable to drug combinations as well as single treatments even for experimental drugs or their combinations that have never been seen in any clinical trial in the labeled data. Moreover, PlaNet captures causal relationships between population variability and treatment outcome, suggesting populations at risk of developing adverse events whose exclusion can impact the outcome of the trials and reduce the likelihood of side effects.

6.2 Results

6.2.1 Overview of PlaNet knowledge graph

PlaNet integrates the treatment information with an underlying biological and chemical knowledge. PlaNet consists of two knowledge graphs (KG): *(i)* a foreground clinical KG, and *(ii)* a background biological KG that captures relevant biology and chemistry. Clinical KG consists of a *(drug, condition, population)* triplets describing drug that is applied, condition or disease that the given population or patient has, and population/patient properties such as gender, age and medical history. Thus, *(drug, disease, population)* triplet defines a core triplet of the clinical KG that describes an application of a drug to a particular population or an individual. We then connect the foreground clinical KG with the background KG that captures underlying biology and chemistry. To create background biological KG, we integrate 9 biological and chemical databases to capture knowledge of disease biology and drug chemistry such as genomic variants associated with human diseases (Piñero et al., 2016; 2019), drug targets (Wishart et al., 2018), physical interactions between human proteins (Ruiz et al., 2021), protein functions (Ashburner et al., 2000), chemical similarities between drugs (Feunang et al., 2016), molecular, cellular and physiological phenotypes of chemicals (Davis et al., 2018) (Fig 6.1b; Supplementary Note 2). In total, PlaNet captures $5,751$ diseases, $14,300$ drugs

augmented with 4,825 drug structural classes, and 17,660 proteins with 28,734 protein functions.

To demonstrate the usage of PlaNet, we instantiate clinical KG on the clinical trials database[1] (Fig 6.1c). We structure the database and represent it in the form of treatment *(drug, condition, population)* triplets by extracting drug-disease-population information from free-text trial protocol description using various named entity recognition approaches (Supplementary Note 1). Drug corresponds to intervention whose effectiveness or safety is investigated in the trial, disease is a condition that is being studied in a trial, and population is defined by eligibility criteria. By structuring the clinical trials database, we avoid natural language bias and allow grounding the structured entities in a background biomedical KG of PlaNet (Fig 6.1d). Overall, the KG is built over 69,595 interventional clinical trials and 205,809 trial arms. It comprises 13,928,443 edges between 330,915 nodes (Supplementary).

6.2.2 Learning general-purpose embeddings using PlaNet

PlaNet learns general-purpose representations (embeddings) of all entities in the KG including clinical entities in the clinical foreground KG, as well as biological and chemical entities defined in the biomedical background KG. The encoder takes a KG as input and for each entity in the graph generates low-dimensional embeddings that preserve information about the graph topology, while capturing heterogeneity of the graph by learning relation-specific transformations that depend on the type of an edge considered. To learn general-purpose embeddings, we perform self-supervised learning by defining an auxiliary task as predicting the existence of an edge between two entities in the KG (Methods). This auxiliary task does not require any labels and enables PlaNet to learn meaningful embeddings from the prior knowledge data.

Pretraining step generates embeddings of every entity in the KG, in total 330,915 entities. We visualize resulting trial arm entities in the two-dimensional UMAP space (McInnes et al., 2018) (Fig. 6.2a). We find that trial arm nodes cluster based on disease groups and trial arms that investigate more similar diseases are embedded next to each other confirming that learnt embeddings are meaningful. For example, mental and nervous system diseases, and cardiovacular and nutritional/metabolic diseases are embedded close to each other. We demonstrate that these embeddings can be used for knowledge graph query answering over the structured clinical trials and biomedical knowledge databases (Supplementary Note 3). For example, one can ask PlaNet to generate all diseases associated with a protein that a particular drug targets, suggesting potential candidates for drug repurposing (Supplementary). By fine-tuning the PlaNet using task-specific annotations, PlaNet is applicable to a variety of downstream tasks. In particular, we next demonstrate PlaNet's ability to reason about efficacy and safety of clinical trials.

[1] https://clinicaltrials.gov

6.2.3 Predicting efficacy of clinical trials using PlaNet

We applied PlaNet to predict efficacy of drugs in the clinical trials database. We focused on predicting a survival endpoint as the most frequently used primary and secondary outcome. We parsed the survival information from the results section of the clinical trials and ensured that a higher value indicates more positive outcome, obtaining $1,307$ labeled trial arms across 625 trials. Given two arms of the same trial testing different drugs, we aimed at predicting which drug will result in more favorable prognosis (Fig. 6.2b). We represent trial arm as a set of study protocol embeddings including arm, drug, disease, primary outcome and eligibility criteria embeddings and fine-tune PlaNet using survival information.

We compared PlaNet to drug-disease-outcome (DDO) model and transformer–based language model BERT pretrained on the PubMed abstracts and full PubMed Central articles (Devlin et al., 2019; Gu et al., 2021) and fine-tuned on clinical trials protocol text information (Supplementary Note 4). PlaNet achieves 0.70 area under receiver operating characteristic curve (AUROC), outperforming the PuBMedBERT model by 15% (Fig. 6.2c). For instance, PlaNet is the only model that correctly predicted higher overall survival of the atezolizumab group compared to docetaxel group in Phase II non-small-cell lung cancer trial (Chalabi et al., 2020) (Supplementary), as well as the outcome of the recently initiated trial which showed that immunomodulatory agent lenalidomide can increase the activity of rituximab and leads to significantly higher progression-free-survival (Leonard et al., 2019) (Supplementary). To further boost PlaNet with a textual knowledge, we developed a joint knowledge- language model (PlaNetLM) that allows joint reasoning over text and KG, allowing the two modalities to interact with each other (Yasunaga et al., 2021; 2022a) (Methods). We observed an additional 5% improvement in the performance in the fused language-KG PlaNetLM model (Fig. 6.2c). The substantial improvements of PlaNet models are not dependent on the evaluation metric (Supplementary).

Given that the number of training examples is limited to clinical trials that reported results (Prayle et al., 2012; Ross et al., 2009), we further tested whether a larger dataset could provide further boosts in the PlaNet's performance. We sampled without replacement our training set to artificially reduce its size and we found that with larger training set size PlaNet substantially improved performance (Fig. 6.2d). This shows that substantial performance improvements can be expected by increasing the training set size even by only a few hundred examples. While PlaNet is able to reason about drug effectiveness, we also investigated whether we can use PlaNet to search for candidate drugs that have a potential to be more effective than an FDA approved drug for a particular disease by creating artifical AI-generated clinical trials (Supplementary Note 6). We focused our question on capecitabine, an FDA approved treatment for metastatic breast cancer (Ershler, 2006). Among 7 top ranked drugs, all drugs have been investigated for breast cancers in isolation or combination with other drugs with a number of ongoing clinical trials, supporting immediate practical applicability of PlaNet in providing insights in potentially effective treatments.

6.2.4 PlaNet predicts outcome of novel drugs

We next questioned whether PlaNet can be applied to new drugs. This ability is crucial to be able to make predictions for experimental drugs that have never been investigated before. To test that, we train the model on 1040 drugs and then apply it to a new set of 224 drugs that have never been applied in any clinical trial seen in the labeled data. Remarkably, we find that PlaNet achieves comparable performance on novel drugs compared to drugs abundantly present in the training set (Fig. 6.2e), demonstrating that PlaNet effectively generalizes to novel drugs, never-before-tested in the clinical trials. Such strong generalization ability is achieved by exploiting similarities between novel drugs and well investigated drugs through their connections in the KG.

When analyzing individual examples, we find that PlaNet predicted with high confidence lower survival of the novel investigational anticancer agent tasisulam-sodium compared to chemotherapy drug paclitaxel even though the model has never seen any labeled example that investigated tasisulam (Fig. 6.2f). In this phase III study conducted on metastatic melanoma patients, tasisulam resulted in 2.6 months lower overall survival and the trial was early terminated due to the possibly tasisulam-related deaths that were identified by the external data monitoring committee (Hamid et al., 2014). PlaNet is also applicable to drug combinations which is a highly non-trivial capability. For example, PlaNet correctly predicted improved progression-free survival (PFS) of combination of dabrafenib and trametinib compared to trametinib alone for melanoma patients without ever seeing any labeled example of trametinib or dabrafenib in the training set (Fig. 6.2g). Combination of these drugs was shown to be superior compared to monotherapy with 3-year PFS 22% with dabrafenib plus trametinib and 12% with trametinib alone (Long et al., 2017) and it was later approved by FDA for melanoma patients with BRAF V600E or V600K mutations.

6.2.5 Predicting safety of clinical trials using PlaNet

We next applied PlaNet to reason about safety of clinical trials by extracting information about side effects of clinical trials from the results section. While previous works used machine learning models to predict adverse events of a drugs and drug combinations (Atias and Sharan, 2011; Liu et al., 2012; Zitnik et al., 2018; Galeano et al., 2020), these prior works neglect the effect of population to which the drug is applied on the occurrence of adverse events. Same drug applied to different populations may have caused different adverse events. To investigate dependence of adverse events on the change of population, we compared the adverse events frequency distributions between trials that apply the same drug to populations suffering from the same disease and trials in which disease is changed. We find that a high percentage of drug-disease combinations have significantly different adverse events frequency distributions when drug is applied to a different population (Supplementary).

We defined safety of a clinical trial with respect to a prior probability that a population suffering from a particular condition will experience an adverse event without any intervention. We use placebo arm to estimate this prior probability and predict if the occurrence of a particular event is

enriched in the intervention arm compared to the placebo arm when no intervention is given to the participants (Methods). We apply PlaNet to two safety prediction tasks: *(i)* predicting occurrence of a serious adverse event, and *(ii)* predicting exact adverse event category defined based on the preferred term in MedDRA hierarchy (Brown et al., 1999) (Fig. 6.3a). On the serious adverse event prediction task, PlaNet achieves a high AUROC score of 0.79 (Fig. 6.3b). Similar performance is observed on non-cancer clinical trials, confirming that the model is not biased to cancer trials that have higher probability of serious adverse events. We next evaluate whether PlaNet can predict the exact adverse event category. PlaNet achieves average 0.85 AUROC score across 554 adverse event categories, retaining high performance across different adverse event categories (Fig. 6.3c). Since many adverse events have a small number of positives, we additionally measure performance using AUPRC score as a function of the number of positives in the training set. For all bins, PlaNet consistently outperforms all baselines (Supplementary). We next assess the generalization ability of the model to predict safety of drugs and diseases that have never been seen during training. Similar to efficacy results, we again find that PlaNet effectively generalizes to novel drugs and diseases, achieving similar performance on novel drugs and diseases compared to drugs/diseases previously seen (Supplementary).

In the real-world setting, one would like to apply PlaNet to predict outcomes of new clinical trials by using historical data for training. To check how applicable is PlaNet in this setting, we use clinical trials data up to June 2017 for training, and then apply PlaNet to predict safety of newer trials that posted results after that date. We find that PlaNet achieves similar performance as when splitting the data by ensuring unique drug-disease pairs (Fig. 6.3d), demonstrating its applicability in the real-world setting in which the model needs to generalize to future trials. Interestingly, we find that PlaNet assigned very high confidence to pneumonia as an adverse event of everolimus given to patients with tuberous sclerosis complex with refractory partial-onset seizures in a phase III trial which is a very rare adverse event of everolimus (Saito et al., 2013) (Fig. 6.3e). However, we find that in this trial pneumonia was reported as a very common adverse event with one patient dying from pneumonia, which was even suspected to be treatment-related (Curatolo et al., 2018). In a phase II trial that investigated lenvatinib safety for thyroid cancer patients, PlaNet correctly assigned highest confidence to uncontrolled hypertension as an adverse event (Fig. 6.3f). Hypertension was indeed later reported as the most frequent adverse event occurring in 80.5% patients (Giani et al., 2021). PlaNet also correctly predicted with high confidence two other adverse events with the highest frequencies: fatigue (58.3%) and diarrhea (36.1%) (Supplementary). Moreover, in three recent COVID-19 trials that investigated efficacy of remdesivir, PlaNet increased the probability of hemorrhage and breathing difficulty in all trials, which have been consistently reported in COVID-19 patients (Sudre et al., 2021; Patell et al., 2020) (Supplementary). The model has never seen examples with COVID-19 or remdesivir drug during model training. In another COVID-19 trial completed in 2021 that investigated the protective role of proxalutamide in COVID-19 infection, PlaNet correctly

increased the probability of gastrointestinal spasm as a side effect (Supplementary), which was reported as the most common treatment emergent adverse event in this trial (McCoy et al., 2021).

6.2.6 Causal reasoning with PlaNet

The fundamental question of trial design and precision medicine is whether we can change population or patient properties to lead to a more favorable outcomes of treatments. To analyze the sensitivity of PlaNet to subtle changes of population terms, we identified all clinical trials that investigate the same drug, study the same disease and have the same primary outcome, but define different inclusion/exclusion criteria and result in a different adverse event (Fig. 6.4a). Given these matched trials, we aim at analyzing whether PlaNet correctly adjusts probability of an adverse event when the population is changed. We count pairs of matched trials as correct or wrong only if the difference between probability of adverse event occurrence is larger then the predefined threshold that we initially set to 0.2. We find that PlaNet correctly adjusted probability in 91% of matched pairs (6575 out of 7261), while wrong adjustments were observed in only 9% of pairs (Fig. 6.4b). With higher probability thresholds PlaNet achieves even higher differences between the correct and wrong predictions: with probability threshold of 0.3 PlaNet has 22 times more correct than wrong probability adjustments, while with 0.4 threshold PlaNet has 90 times more correct adjustments (Fig. 6.4c).

We next develop a methodology for assigning node importance scores to each term in the eligibility criteria (Methods). Given a population term, *i.e.*, inclusion/exclusion term in case of clinical trials, PlaNet computes the change in adverse event probability when the term is removed from the inclusion or exclusion criteria. High score indicates that removing a term from the criteria has a high influence on the occurrence of an adverse event. We then rank terms based on their influence on adverse event probability change (Fig. 6.4d). Using this methodology, we find that in a trial that investigated efficacy and safety of exemestane for breast neoplasms, PlaNet indicates that excluding terms 'metastasis', 'exemestane', 'tamoxifen' and 'aromatase inhibitors' leads to the lower probability of breathing difficulty (Fig. 6.4e). We validate this finding by identifying another related trial that also studied exemestane for breast neoplasms but it does not have these terms in the exclusion criteria, being focused on metastatic breast neoplasms. Indeed, breathing difficulty is significantly enriched in a metastatic breast cancer trial compared to placebo and comparing PlaNet's predictions between these two trials PlaNet correctly adjusted probabilities and assigned 21.8% higher probability of breathing difficulty for the metastatic breast neoplasm trial. Additionally, external validation in literature and drug reports confirms that breathing difficulty is a known symptom of metastatic breast cancer (Geels et al., 2000) and a potential adverse event of tamoxifen and aromatase inhibitors including exemestane (Peters and Tadi, 2021).

6.3 Discussion

PlaNet is a geometric deep learning framework for predicting treatment response of a population by reasoning over a massive clinical knowledge graph. The clinical knowledge graph in PlaNet captures population heterogeneity and prior knowledge of biological and chemical interactions. PlaNet learns low-dimensional embeddings of heterogeneous node types in an unsupervised manner and can use them on downstream pharmacological tasks of interest, such as predicting drug efficacy and likelihood of serious adverse events. If text data is additionally available, PlaNet can be further complemented with the language models (Devlin et al., 2019; Lee et al., 2020) and trained as a joint knowledge-language foundation model (Yasunaga et al., 2022a).

The unique ability of PlaNet is its ability to generalize to drugs, diseases and population terms that have never been part of the annotated datasets. By modelling clinical terms as nodes in the massive knowledge graph, PlaNet finds similarity of the novel terms to existing terms. This enables PlaNet to make predictions for experimental drugs, new emerging disease states, or population properties that have not been tested before. In three COVID-19 trials that investigated efficacy of remdesivir – disease and drug for which PlaNet has never seen any annotated example – PlaNet increased the probability of hemorrhage and breathing difficulty, side effects that have been consistently reported in COVID-19 patients (Sudre et al., 2021; Patell et al., 2020). While previous works showed advantage in using network-based methods to identify clinically efficacious drug combinations (Cheng et al., 2019), PlaNet extends this capability not only by considering population heterogeneity, but also by making predictions for combinations that include novel, experimental drugs.

PlaNet is scalable, flexible and easily extendable. Without retraining the model, PlaNet can be applied to new entities in the treatment knowledge graph such as new drugs, new diseases and new population terms. This important feature allows obtaining predictions for new drugs and population properties without retraining the model on these new terms.

PlaNet is uniquely able to reason about treatment effects over a complex population space and suggest how to change population to reduce the negative effects of the treatment. This opens opportunities to design more safe and effective treatments by intervening in the population design, but also to discover interventions effective only in certain groups. So far, such discovery has been happening rarely and often by chance (Schork, 2015).

Finally, PlaNet is a general framework: although we demonstrate its usage on clinical trials data, it could also be used to represent individual patients and integrated with existing clinical knowledge graphs (Santos et al., 2022). In that case, the population properties would correspond to individual patient characteristics such as personal omics assays (Chen et al., 2012) paving the way towards precision medicine (Ashley, 2016).

6.4 Method

6.4.1 Knowledge graph construction

We develop a computational framework for systematically extracting structured information from the clinical trials database.[2]

We focus on interventional clinical trials that study at least one drug, resulting in 69,595 trials. Given free-form text description of a clinical trial, our framework automatically extracts and structures clinical trials protocol information, including disease, drug/intervention, primary outcome and eligibility criteria. After extracting key terms, we standardize them by mapping the extracted terms to external databases. We provide details of the knowledge graph construction pipeline in Supplementary Note 1.

6.4.2 Model Overview

PlaNet knowledge graph is represented as a directed and labeled multi-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R}, \mathcal{T})$ where $v_i \in \mathcal{V}$ are nodes/entities, $(v_i, r, v_j) \in \mathcal{E}$ are relations/labeled edges, $t_i \in \mathcal{T}$ are node types and $r \in \mathcal{R}$ denote relation types. Additionally, entities have associated entity attributes depending on the entity type (Supplementary Note 1). PlaNet learns a low-dimensional representation z_i for all the entities in the graph $\mathcal{G}$. The low-dimensional entity representations are learnt to capture both structural properties of an entity's neighborhood as well as entity's attribute representations.

6.4.3 Encoder

The encoder model takes node/entity in the PlaNet and maps it to a low-dimensional embedding vector that captures entity attributes and its local neighborhood. Formally, the encoder is a function $ENC : \mathcal{V} \rightarrow \mathbb{R}^d$ that takes entity $v_i \in \mathcal{V}$ and generates its low-dimensional embedding $z_i \in \mathbb{R}^d$ that captures entity structural properties as well as entity attributes. We build our encoder model as the relational graph neural networks (R-GCN) (Schlichtkrull et al., 2018) encoder. Given a latent low-dimensional representation $h_i^{(l)}$ of entity v_i in the l-th layer of the neural network, single layer of the encoder has the following form:

$$h_i^{(l+1)} = \sigma\left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)}\right), \tag{6.1}$$

where $W_r^{(l)}$ is the transformation matrix for relation $r \in \mathcal{R}$, $\mathcal{N}_i^r$ denotes the set of neighbor indices of node i under relation $r \in \mathcal{R}$, $c_{i,r}$ denotes normalization constant defined as $c_{i,r} = |\mathcal{N}_i^r|$ and the operator σ defines the non-linear function in the neural network model. We use PReLU as an

[2] https://clinicaltrials.gov

activation function. The key idea of the relational encoder is to learn propagation and transformation operators across different parts of the graph defined by the entity and relation types. Since the transformation matrix depends on the relation type, the encoder propagates latent node feature information across edges of the graph while taking into account the type of an edge. In this way local neighborhoods are accumulated differently depending on the entity type. Thus, for each entity in the graph the encoder has a different neural network architecture defined by the network neighborhood of the given entity.

In the first layer, $h_i^{(0)}$ is initialized with entity attributes. The entity feature vectors are associated with different entity types, so we first learn a linear projection W_{t_i} for each entity type $t_i \in \mathcal{T}$ and use the projected attributes as the input to the first layer of the network:

$$h_i^{(0)} = W_{t_i} x_i \tag{6.2}$$

where x_i is an entity of type t_i. In other layers, the output of the previous layer becomes the input to the next layer representing latent low-dimensional entity representations that capture neighborhood structure. Stacking multiple layers allows successive application of propagation/transformation operators, giving the ability to the model to capture higher-order network neighborhoods. Final representation of entity v_i in the last (L-th) layer of the encoder gives us entity embeddings $z_i \in \mathbb{R}^d$, that is $ENC(v_i) = z_i = h_i^{(L)}$.

To efficiently handle rapid growth in the number of parameters with the number of relations in the graph, we use the basis decomposition regularization technique (Schlichtkrull et al., 2018) and represent transformation matrix as a linear combination of basis transformations:

$$W_r^{(l)} = \sum_{b=1}^{B} a_{rb}^{(l)} V_b^{(l)}, \tag{6.3}$$

where $V_b^{(l)} \in \mathbb{R}^{d^{(l+1)} \times d^{(l)}}$ define basis and $a_{rb}^{(l)}$ are coefficients that depend on relation r.

6.4.4 Self-supervised learning

To leverage a large amount of unlabeled data, we first perform self-supervised learning by defining an auxiliary task. We define the auxiliary task as the edge mask/link prediction task. In particular, for each triplet (h, r, t) consisting of head, relation and tail entities, we first construct a k-hop subgraph of the head and tail entities. Then, we randomly drop α edges in the subgraph and the model is asked to reconstruct the dropped edges by assigning scores $f(h, r, t)$ to possible edges (h, r, t) in order to determine how likely those edges belong to $\mathcal{E}$. Our model for the task is a graph auto-encoder model, consisting of an entity encoder and an edge scoring function as the decoder. The encoder maps each entity $v_i \in \mathcal{V}$ to a real-valued vector $z_i \in \mathbb{R}^d$. The decoder assigns scores to (h, r, t)-triplets through a scoring function $f : \mathbb{R}^d \times \mathcal{R} \times \mathbb{R}^d \rightarrow \mathbb{R}$ denoting the probability of the triplet belonging to the

graph. To define the scoring function for the triplets, we use DistMult factorization decoder (Yang et al., 2015):

$$f(h, r, t) = z_h^T R_r z_t \,. \tag{6.4}$$

where every relation r is associated with a diagonal matrix $R_r \in \mathbb{R}^{d\times d}$, while z_h and z_t denote head and tail embeddings, respectively. We train the model with negative sampling (Yang et al., 2015; Schlichtkrull et al., 2018) meaning that for each observed example we sample n negative edges by randomly corrupting either the head or the tail of each positive triplet but not both. We use negative sampling loss with self-adversarial negative sampling (Sun et al., 2019b) as defined below:

$$L = -\log\sigma(f(h, r, t)) - \sum_{i=1}^{n} p(h'_i, r, t'_i)\log\sigma(-f(h'_i, r, t'_i))\,, \tag{6.5}$$

with

$$p(h'_j, r, t'_j|\{(h_i, r, t_i)\}) = \frac{e^{\alpha f(h'_j, r, t'_j)}}{\sum_i e^{\alpha f(h'_i, r, t'_i)}}\,, \tag{6.6}$$

where α is the sampling temperature, σ is the sigmoid function, and (h'_i, r, t'_i) is the i-th corrupted triplet for the positive triplet (h_i, r, t_i).

6.4.5 Outcome prediction

To fine-tune PlaNet on downstream prediction tasks, we represent a trial arm as the set of entities defining the trial protocol information including trial arm, diseases, drugs, primary outcomes, included and excluded population. To obtain trial arm embedding, we first obtain a representation vector for each entity type of trial protocol entity by computing the average embedding of all entities of a given type. Resulting embeddings represent protocol embeddings, *i.e.*, drug embedding, disease embedding, included/excluded population embeddings and primary outcome embedding. Finally, we concatenate all entity embeddings including arm embedding to obtain final trial representation h_T. Formally, the final trial arm embedding h_T is computed by aggregating information from all protocol entities using a parameter free convolution layer:

$$h_T = \left(\|_{r\in\mathcal{R}_T}\frac{1}{|\mathcal{N}_i^r|}\sum_{j\in\mathcal{N}_i^r} h_j^{(L)}\right) \| \, h_T^{(L)} \tag{6.7}$$

where R_T denotes relations of a trial arm and $h_T^{(L)}$ is trial arm representation in the last layer L.

Trial outcome classifier takes as input final trial arm embedding and predicts the outcomes of the clinical trials, namely efficacy, safety and exact adverse events category. For efficacy prediction, outcome classifier takes as input pair of trial arm embeddings, while for safety and efficacy tasks classifier takes as input single trial arm embedding. Task-specific classifier consists of two fully connected layers and outputs the probability that a particular event occurs. Specifically, the trial

encoder is followed by a fully connected layer with non-linear ReLU activation function. Given trial arm embedding h_T, the forward-pass update of the first fully connected classifier layer is the following:

$$h'_T = ReLU(W_{T'} h_T + b_{T'}), \tag{6.8}$$

where $W_{T'}$ is a parameter matrix and $b_{T'}$ is a bias vector. Finally, the model outputs probabilities in the second layer:

$$p = \sigma(W_t h'_T + b_t), \tag{6.9}$$

where W_t is the task specific weight matrix, b_t is the task specific scalar bias, and σ is the logistic sigmoid function.

6.4.6 Efficacy prediction

In the efficacy task, we predict which arm will have more favorable outcomes. We consider only survival-related primary and secondary outcomes including overall survival, progression-free survival, recurrence-free survival and disease-free survival. Depending on the unit, higher value may indicate better or worse outcome and we correct all examples with the opposite direction. The output of the model represents the probability that the first arm will have higher survival than the second arm. Specifically, given a pair of arms, we concatenate their trial arm embeddings computed from Equation (6.7), and then apply Equations (6.8) and (6.9) for prediction. We use the binary cross-entropy loss for training.

6.4.7 Safety and adverse event prediction

In the safety prediction task, the output corresponds to the probability of occurrence of serious adverse events, while in the adverse event prediction task the output corresponds to the probability of the occurrence of a particular adverse event category. We define both tasks with respect to the placebo arm. The placebo arm represents the prior probability that the adverse event will occur given the disease and population that the clinical trial is investigating. For each disease, we aggregate information from all tested placebo arms and use it as the estimation of the expected safety issues/adverse events. Given an intervention, we then construct a contingency table of frequency distributions between treatment and the estimated placebo arm and check whether the enrichment of adverse events is higher in the treatment arm than in the placebo arm at the particular odds ratio threshold. We use the odds ratio 2 as the default threshold. Importantly, the frequency between true placebo arms and estimated placebo arms is not significantly different between true and estimated placebo arms in 99.4% trials (t-test, , FDR $< 10\%$), confirming that our estimates are trustworthy.

For predicting adverse events we consider MedDRA Primary Term (PT) level terms with at least 50 positive examples and at least 15 positive examples in the test set. In the adverse events prediction task, many categories are scarcely labeled. To transfer useful information from abundantly labeled

categories to scarcely labeled categories, we train our model in the multi-task setting, In particular, our loss function is a multi-task binary cross entropy loss:

$$\mathcal{L}_{AE} = -\sum_{c \in C} \frac{1}{N_c} \sum_{j=1}^{N_c} y_{jc} \log p_{jc} + (1 - y_{jc}) \log(1 - p_{jc}), \tag{6.10}$$

where C is the set of adverse event categories, N_c is the number of learning examples for category task c, y denotes outcome binary labels and p denotes probability at the output of the model defined in Equation (6.9). Encoder is shared across all tasks, while each task has its own task-specific classifier. In particular, classifier parameters in Equation (6.8) are shared across all tasks, while parameters in Equation (6.9) are task-specific. For the safety prediction task, we use binary cross-entropy loss. We split the data into train, validation and test sets by ensuring that same trial and same drug-disease pairs can not appear in different splits, meaning that the model needs to generalize to unseen drug-disease combinations.

6.4.8 Knowledge graph-language model framework (PlaNetLM)

The PlaNet model discussed above uses our constructed PlaNet knowledge graph as the primary information for efficacy/safety prediction. In addition, the raw text of clinical trial protocols could provide additional context (*e.g.*, description of the exact way dosage is given to participants), and improve robustness and safety of the model. With this motivation, we introduce a version of PlaNet model that incorporates the textual information (PlaNetLM), where we augment the R-GCN encoder with a text encoder, inspired by the DRAGON method (Yasunaga et al., 2022b;a). Specifically, letting $\texttt{text}_T$ denote the protocol text of the input trial arm T, we use a Transformer encoder (Vaswani et al., 2017) to obtain a text embedding of the arm, $g_T = \text{Transformer}(\texttt{text}_T)$. We then fuse the R-GCN embedding of the arm h_T and the text embedding of the arm g_T by concatenating them and passing them to an MLP. We use this architecture for both the pre-training and fine-tuning phases.

6.4.9 Neural network architecture

Our encoder consists of 2 message passing layers with 512 embedding size in each layer and basis decomposition with 15 bases. We use layer normalization, and PReLU (He et al., 2015) activation after the first layer of message passing. Additionally we use a Dropout (Srivastava et al., 2014) of 0.2 for the encoder after each layer. Other parameters are reported in Supplementary Note 5.

6.4.10 Causal reasoning

To provide explanations behind the predictions for the input trial arm, we develop a methodology for assigning node influence scores to each term in the eligibility criteria, inspired by (Ying et al.,

2019). Given a term, we compute the change in adverse event probability when the term is removed from the inclusion or exclusion criteria. Concretely, denoting the input trial arm node as T, the eligibility criterion term node as e, and the TrialNet KG as G, we prepare a KG *without* the edges between T and e: $G' = G \backslash \{(e, T)\}$. Then the influence score of the eligibility criterion e for the trial arm T in the adverse event category c is computed as:

$$S_c^{e \to T} := \Delta p_c = p(y_c; G') - p(y_c; G) \tag{6.11}$$

If the score is positive, it indicates that removing this eligibility criterion makes the adverse event probability higher, meaning that having this eligibility criterion reduces the adverse event probability.

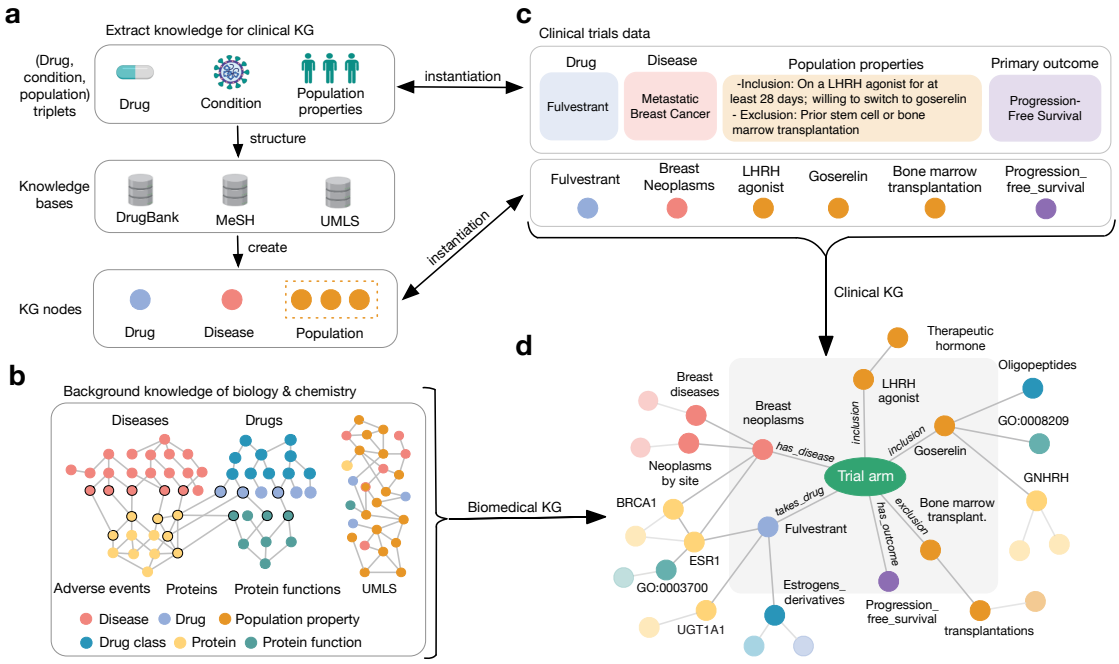


Figure 6.1: **Overview of the PlaNet framework.** PlaNet is built as a massive clinical knowledge graph (KG) that captures treatment information as well as underlying biology and chemistry. **(a)** The core of the PlaNet framework is a clinical KG that represents knowledge in the form of *(drug, disease, population)* triplets. These entities are then linked to external knowledge bases: diseases to Medical Subject Headings (MeSH) vocabulary (Lipscomb, 2000), treatments to DrugBank database (Wishart et al., 2018), and population properties to Unified Medical Language System (UMLS) terms (Bodenreider, 2004). **(b)** We integrate 11 biological and chemical databases to capture knowledge of disease biology and drug chemistry, such as databases of drug structural similarities, drug targets, disease-perturbed proteins, protein interactions and protein functional relations (Methods). These databases are integrated with the UMLS graph that captures population relations. **(c)** Instantiation of the PlaNet framework on the clinical trials data. We parse and standardize clinical trials database and extract information about diseases, drug treatments, eligibility criteria terms and primary outcomes. **(d)** Final KG is obtained by integrating the clinical KG (c) with biological and chemical networks (b).

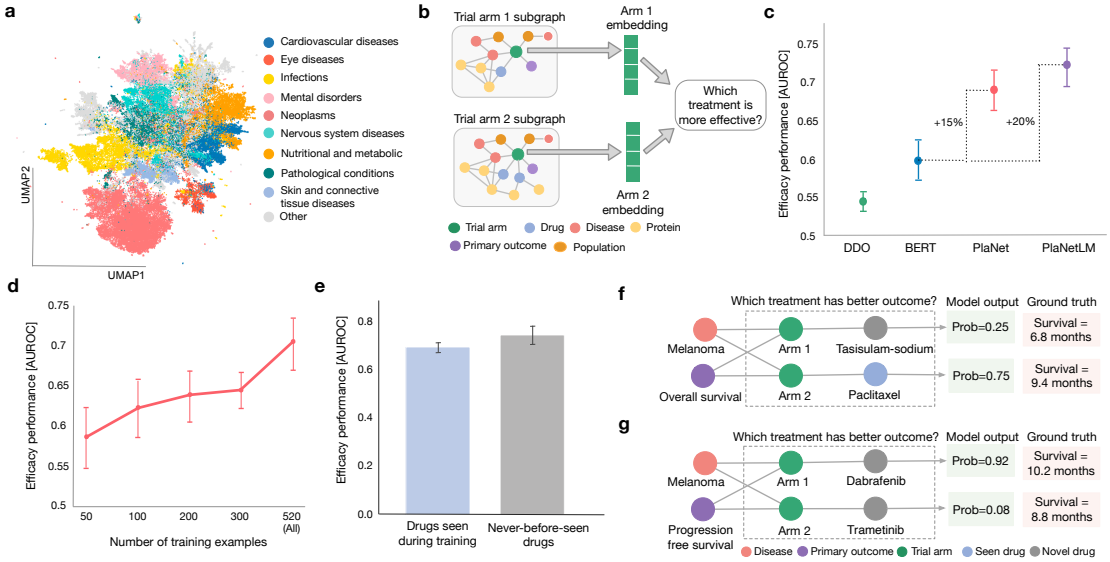


Figure 6.2: **PlaNet reasons about efficacy of drugs in clinical trials even for experimental drugs that have never been tested before.** **(a)** UMAP space of all trial arm embeddings in the clinical trials database obtained by pretraining PlaNet on the self-supervised task (Methods). Arms are colored according to disease information. Only major disease groups according to MeSH hierarchy (Lipscomb, 2000) are shown. Grey color denotes minor disease groups. The arm embeddings learned by PlaNet exhibit clustering according to disease groups. **(b)** Given embeddings of two trial arms to which different drug treatments were applied, PlaNet predicts which of the treatments is more effective. Methodologically, the method geometric deep learning model is fine-tuned on the efficacy prediction task by using information about drug efficacy from the completed clinical trials. **(c)** Performance comparison of PlaNet with disease-drug-outcome (DDO) classifier and transformer-based language model BERT (Devlin et al., 2019; Gu et al., 2021). PlaNetLM is obtained by augmenting PlaNet with the text embedding of the trial arm protocol (Yasunaga et al., 2022a) (Methods). Performance is measured as the mean area under receiver operating characteristic curve (AUROC) score across 10 runs of each model on different test data samples. Error bars are 95% bootstrap confidence intervals. **(d)** Effect of the training set size on the performance. With more training data, PlaNet substantially improves performance strongly indicating that further improvements can be expected by increasing the size of the training set. Performance is measured as the mean AUROC score across 10 runs on different test data samples. Error bars are 95% bootstrap confidence intervals. **(e)** PlaNet predicts efficacy of novel, experimental drugs that have never been seen in a clinical trial before. Bars represent the mean AUROC score for drugs that have been seen in the labeled training data (left; blue color), and never-before-seen drugs (right; grey color). Mean performance is computed across 10 runs of different test data samples and error bars are 95% bootstrap confidence intervals. **(f, g)** Examples of correct predictions. PlaNet outputs probabilities that a particular treatment will lead to higher overall survival of the population. **(f)** PlaNet correctly predicted higher overall survival of melanoma patients in paclitaxel arm compared to tasisulam-sodium arm. The model has never before seen any effect (labeled example) of the tasisulam-sodium drug. **(g)** PlaNet correctly predicted higher progression free survival of melanoma patients when given combination of dabrafenib and trametinib drugs compared to trametinib drug alone. The model has never before seen any effect of dabrafenib or trametinib drugs.

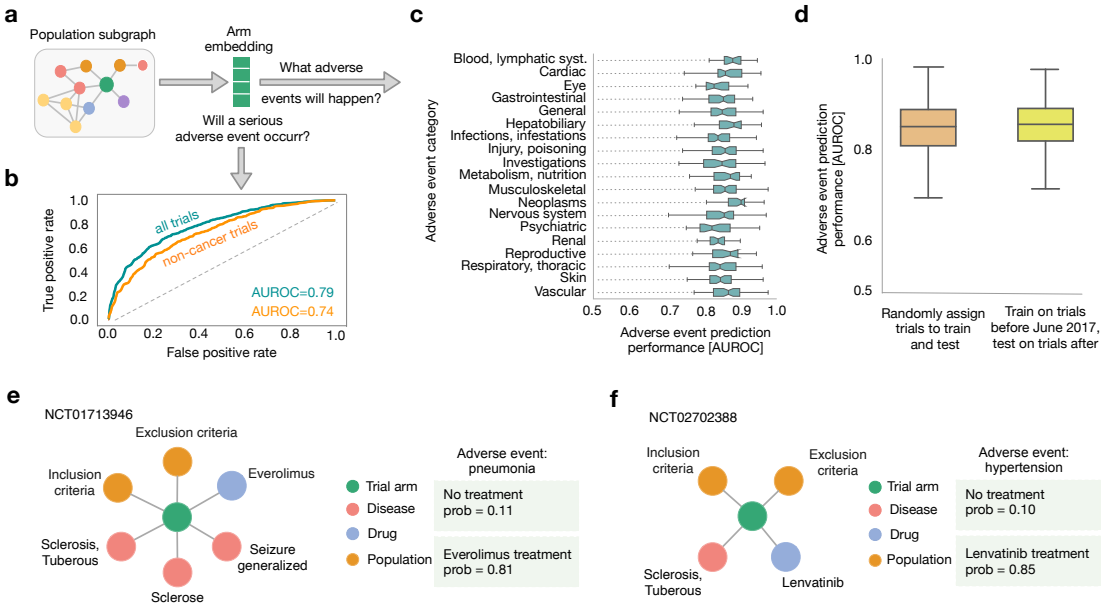


Figure 6.3: **PlaNet reasons about safety of clinical trials.** **(a)** Given a trial arm embedding, PlaNet predicts (b) whether a serious adverse event will occur and (c) what adverse event will happen. Methodologically, the methodolog geometric deep learning model is fine-tuned on the safety task by using information about drug safety from the completed clinical trials. **(b)** Performance of PlaNet on predicting occurrence of serious adverse events. PlaNet achieves AUROC score of 0.79 on predicting whether serious adverse event will occur. Green curve shows performance on all trials, while orange curve shows performance on on trials that do not investigate cancer diseases. **(c)** Performance of PlaNet on predicting exact category of adverse events measured as AUROC score. We consider 554 adverse events defined as preferred terms (PT) in MedDRA hierarchy (Brown et al., 1999) and group them according to the organ level categories. We consider organ level categories with at least 20 PT terms. The boxes show the quartiles of the performance distribution across different adverse events. Whiskers show the rest of the distribution. **(d)** Performance of PlaNet on predicting adverse events of future clinical trials. PlaNet achieves similar performance on predicting outcome of future clinical trials when compared to trials that are randomly split into train and test dataset independent of the year in which they were conducted. The performance is measured using AUROC and boxes show quartiles of the AUROC distribution across different adverse events. Whiskers show the rest of the distribution. **(e, f)** Examples of individual predictions of adverse events. Model assigns probability that an adverse event will be enriched in a given arm compared to no-treatment arm (Methods). **(e)** In an everolimus safety trial for tuberous sclerosis complex with refractory partial-onset seizures, PlaNet correctly predicted pneumonia as an adverse event with a high confidence. Although pneumonia is a very rare adverse event of everolimus (Saito et al., 2013), in this trial pneumonia was reported as a very common adverse event with one patient dying from pneumonia, which was suspected to be treatment-related (Curatolo et al., 2018). **(f)** In a lenvatinib safety trial for thyroid cancer patients, PlaNet correctly predicted uncontrolled hypertension as an adverse event. Uncontrolled hypertension was reported as the most frequent adverse event in that trial (Giani et al., 2021).

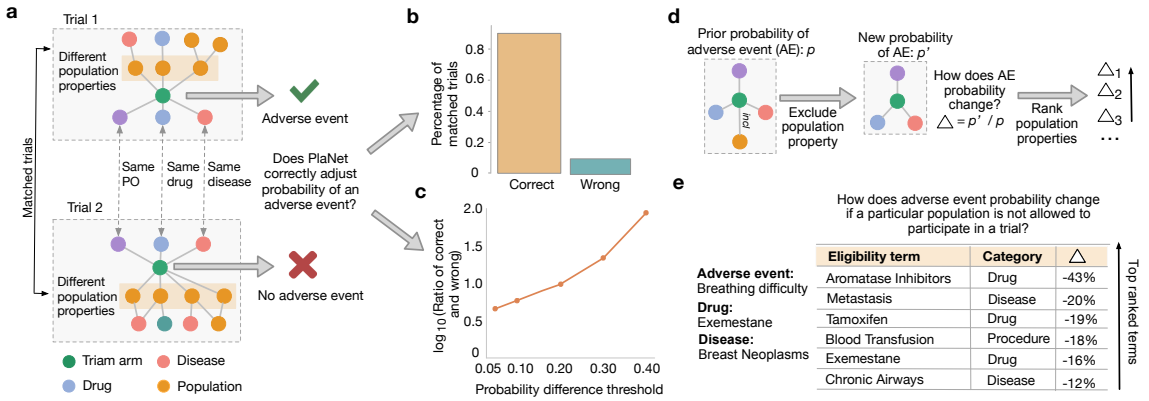


Figure 6.4: **PlaNet identifies characteristics of populations that are at risk of developing adverse events.** **(a)** We match clinical trials that study same drug, same disease and have same primary outcome (PO), but differ in the characteristics of the eligible population and result in different adverse events, *i.e.*, adverse event was observed in one trial, but not in the other. For pairs of such clinical trials, we assess whether model correctly adjusted prediction of an adverse event and predicted higher probability of an adverse event in one trial compared to the other. **(b)** Percentage of matched trials on which PlaNet correctly adjusted the probability of an adverse event (orange color; left) and percentage on which the adjustment was wrong (green color; right). PlaNet makes 10 times more correct adjustments than wrong. We count pairs only if the difference between probability of adverse event occurrence of two matched trials is at least 0.2. **(c)** The effect of the probability difference threshold on the ratio of correct and wrong probability adjustments. Even with smaller difference in probabilities (at least 0.05), the number of correct adjustments is more than 4 times higher than the number of wrong adjustments. With the difference of at least 0.4 the number of correct adjustments is 90 times higher than the number of wrong adjustments. For each probability threshold p, we count matched trials as correct or wrong only if the difference between probabilities is at least p. **(d)** PlaNet identifies population characteristics whose exclusion can reduce probability of adverse events. Given a population property, we estimate prior probability of an adverse event when population with a given property is included in the trial. We then change the trial by excluding population with that property, and observe the change in adverse event probability Δ. By ranking terms according to probability score, we can identify population properties whose exclusion can increase safety of clinical trials. **(e)** Use case of (d) for a trial that tests exemestane drug for breast neoplasms and in which breathing difficulty was observed as an adverse event. PlaNet finds population properties that have the highest effect on causing breathing difficulty. By excluding that population from the trial, PlaNet suggests that the probability of breathing difficulty can be significantly reduced. We rank terms that belong to drug, disease and procedure categories.

6.5 Supplementary

Here we present details about datasets used to evaluate PlaNet's performance, and information about experimental setup of baseline methods.

6.5.1 Constructing knowledge graph from clinical trials database

We develop a computational framework for systematically extracting structured information from the clinical trials database.[3] We downloaded a snapshot of the database on February 14th 2021.

Extracting arm information Each interventional clinical trial consists of a single or multiple arms defined as the group of participants that receive specific treatment, or are part of the control group which is often placebo group that does not receive any intervention. Arms within the same trial differ from each other in the treatment information, while sharing all other protocol information (disease, eligibility criteria, primary outcomes). We represent a trial as a set of trial arms, where each arm is associated with the drugs given to the participants in that arm. We extract arm information from the trial text and associate the arms to corresponding drugs. We filter out arms which contain any non-drug intervention.

Extracting drug and dosage information Information about the drugs, their dosages, and duration of the treatment is given in the non-standard format and it can appear at different parts of text including intervention name, intervention description, arm title and arm description. Our framework extracts drug names and dosages present in any of these sections and it is built upon MedEx tool (Xu et al., 2010) which was originally developed to extract medication information from clinical notes. We adapt the tool to clinical trials data by extending its lexicon and incorporating additional grammar rules based on manual analysis of the parsing results. Many trials do not report dosage information mostly because the standard dosage is given to the participants. To distinguish between trials that do not report dosage and trials for which MedEx does not capture information, we use BioBERT (Lee et al., 2020) trained on n2c2 2018 challenge data (Henry et al., 2020) to recognize entities such as drug, strength, duration, frequency, route, and form. If no dosage information is detected, we substitute the standard dosage from DrugBank (Wishart et al., 2018) for the corresponding drug applied to the corresponding disease. Additionally, the system recognizes the relationships between drugs and other named entities and we use it to extract the drug dosage information missed by our extended MedEx system. We further normalize the units of the strength of the drug dosage and normalize the frequencies using the standard TIMEX notation (Pustejovsky et al., 2003).

Once drug information is extracted, we standardize drug information by mapping it to the DrugBank database (Wishart et al., 2018). We construct a dictionary of drug names using primary

[3] https://clinicaltrials.gov

names, synonyms, international brands, products, mixtures, and external identifiers of drug entries provided in the DrugBank. We additionally utilize synonyms of the PubChem compound (Kim et al., 2021b), RxNorm identifiers (Nelson et al., 2011) and multi-vocabulary mappings provided in UMLS Metathesaurus (Bodenreider, 2004). We create a dummy drug to represent placebo and construct mappings from different terms used to describe placebo to the dummy drug. Using this approach, we map drug names in 86% of the total arms. Finally, we consider only arms for which all drug names have been extracted, resulting in 205,809 arms from 69,595 clinical trials.

Extracting disease information The condition/disease investigated by the clinical trial is given as a free text and optionally as a set of Medical Subject Headings (MeSH) terms (Lipscomb, 2000). In MeSH, we use the Diseases (C) and Psychology (F) branches to match the disease listed in a clinical trial to a MeSH heading in the tree. The MeSH terms in the clinical trials often include all terms in the MeSH hierarchy up to the most general parent. To select only the most relevant terms, we keep the most specific child MeSH terms as specified by the MeSH hierarchy. We mapped the disease information to MeSH heading for 94% trials. Examples of unmapped conditions include terms such as kidney transplantation, anesthesia, colonoscopy preparation, aging.

Extracting primary outcomes Primary outcomes are defined in the clinical trials as a short unstructured text. We constructed our own controlled vocabulary of primary outcome measures to structure this data and extract relevant information. We first detect common phrases in the primary outcomes across all clinical trials based on the unigram and bigram counts. Given two words w_i and w_j, bigram score is defined as:

$$score(w_i, w_j) = \frac{count(w_i w_j) - \delta}{count(w_i) \times count(w_j)}, \tag{6.12}$$

where *count* defines number of occurrences of bigram or unigram and δ is a parameter that prevents phrases of very infrequent words to be formed. We set δ of 5 and consider only bigrams with a score above the chosen threshold of 5 as phrases. We run another pass over the training data with the same threshold value, allowing longer phrases that consist of several words to be formed. Threshold parameters are chosen based on manual analysis of the resulting phrases. To reduce the noise and account for variation in spelling and terminology, we clustered all extracted phrases. We used agglomerative average clustering using Jaro-Winkler similarity (Winkler, 1990). We considered two phrases to be similar if the similarity is greater than 0.85. Finally, we obtained a vocabulary of 3,048 terms represented by the phrase clusters. We then mapped primary outcomes of all interventional clinical trials data to the clusters based on the phrase present in the outcome text.

Extracting eligibility criteria The eligibility criteria (EC) defines participants eligible to apply for the clinical trials. It consists of inclusion and exclusion criteria. Inclusion criteria defines the

population that can participate in the clinical trial, while exclusion criteria defines population that is not allowed to participate in the clinical trials. Eligibility criteria is specified in the form of the free text as a set of bullet points for both inclusion and exclusion criteria. Our pipeline for structuring and extracting relevant information from the eligibility criteria consists of two parts: (i) entity extraction and (ii) entity linking to the UMLS vocabulary (Bodenreider, 2004).

Entity extraction To extract named entities from the free-text eligibility criteria description, we rely on the Criteria2Query (Yuan et al., 2019) which combines machine learning and rule-based methods to systematically parse eligibility criteria text. The entities extracted from the eligibility criteria text are then mapped to the UMLS metathesaurus (Bodenreider, 2004) by tf-idf (Sammut and Webb, 2010) based text matching. Due to the variability and non-standard nature of the eligibility criteria text, the extracted concepts are sparse, *i.e.*, occur in just a few clinical trials. To address this issue, we exploit the hierarchical structure of the UMLS knowledge graph and map low-frequency concepts to their parents, ensuring that all concepts occur at least in 10 trials in our dataset. To avoid losing information due to dropping low-frequency concepts, we keep the grandparent node of a concept even if both parent and grandparent have a frequency of less than 10. Finally, we map $273,357$ out of the $352,367$ (77.58%) extracted entities and obtain $32,851$ unique UMLS concepts.

Adverse events extraction Adverse events that occurred in the clinical trial are reported in the Results section and grouped into serious and other adverse events. Out of the $69,595$ interventional trials considered in this work, $23,238$ have results published with the trials. We map adverse events to the Lowest Level Terms (LLT) of the MedDRA hierarchy (Brown et al., 1999) using UMLS REST API search and tf-idf matching. Since the LLT is very broad and contains different forms of the same concept, we map all LLT terms to the Preferred Term (PT) for use in our work. To match reported adverse event frequencies to the trial arms, we extracted drug and dosage details from the result group title and description using the same approach as described in drug and dosage paragraph and match if drug and dosage are the same. We additionally incorporate a variety of hand-designed rules for mapping arms developed by investigating unmapped arms. For result groups that are left unmapped but have all drug and dosage details extracted, we create a trial arm corresponding to these result groups and include them in our dataset. In total, we extract information for 30,970 result arms.

Entity attributes We introduce different entity attributes depending on the entity type. Diseases, drugs, primary outcomes and trial arms are embedded as text descriptions using PubMedBERT (Gu et al., 2021) and BioLinkBERT (Yasunaga et al., 2022b). In particular, diseases are represented as descriptions of the MeSH terms, drugs as drug DrugBank descriptions, primary outcomes as concatenated phrases representing the outcome cluster, and trial arms as a trial text description

including brief summary, arm information, intervention details, primary outcome measures and the eligibility criteria. For trial arms, we additionally include a feature vector that represents trial structured information: phase, enrollment, maximum and minimum age of eligible participants and the eligible sex of the participants. For adverse event prediction task, we also represent adverse events as PubMedBERT embeddings. Protein attributes are obtained as elementary biophysical features with a set of engineered representation of proteins (Ofer and Linial, 2015). Population attributes are embedded with *cui2vec* (Beam et al., 2019) UMLS embeddings learned using a large collection of multi-modal medical data based on scientific articles and insurance claims. Drug classes and protein functions are initialized as one-hot encodings. Since features are entity type-specific, they have different dimensionality. To map them to the joint embedding space, all feature vectors are followed by a linear transformation and learnt jointly with the encoder model.

6.5.2 Constructing background knowledge graph about chemistry and biology

Incorporating chemistry and biology background knowledge We construct a knowledge graph to represent biological and chemical knowledge and integrate it with the trials knowledge graph described in the Supplementary Note 1. We construct drug hierarchy and drug-drug network to capture the chemical knowledge and relationships between different drugs. We construct several networks, including disease, protein, function, and population networks, to represent the biological similarities between these entities. We further add several cross-networks like drug-protein and disease-protein to encode these entities' different interactions with one another. All networks are detailed below.

Disease-Disease Network To represent relations between different diseases and construct a disease-disease network, we utilized a subset of the MeSH (Medical Subject Headings) (Lipscomb, 2000) medical concept hierarchy. Out of the total 16 top-level categories in MeSH, we selected the following subset of the hierarchy representing the diseases: C (Diseases), F01 (Behavioral diseases), and F02 (Psychological diseases). We additionally utilized the identity mapping provided by UMLS to merge identical disease nodes in this sub-network. The resulting network is hierarchically organized, and it comprises $5,751$ disease nodes with $17,510$ edges between them represented with 'is-a' relationship between the diseases.

Drug Chemical Hierarchy The relations between drugs are based on the similarity of the drug chemical structure. We construct a drug-drug network using a hierarchical chemical classification of drugs from ClassyFire (Feunang et al., 2016). ClassyFire is based on the chemical taxonomy named ChemOnt, which covers $4,825$ chemical classes of organic and inorganic compounds. The chemical taxonomy has a tree structure representing chemical classes of different granularity and

consists of 4,824 edges between the chemical classes. Furthermore, DrugBank (Wishart et al., 2018) provides the chemical classes of drugs. Based on the chemical structure and properties of the drug, these DrugBank classes relate a drug to multiple chemical classes in the ChemOnt taxonomy at the most fine-grained level. We joined the ChemOnt tree with the drug-drug class relationships from DrugBank to obtain a drug hierarchy to construct a drug-drug network representing structural relationships. The final network consists of 14,300 drugs, 00 drug classes, 8,024 drug class hierarchy relations, and 133,661 drug-drug class relations.

Biological Function Network We construct a hierarchy of biological functions by using the Gene Ontology (GO) (Ashburner et al., 2000; Consortium, 2021a). The Gene Ontology represents a curated hierarchy of biological functions that describe the molecular activities of genes. We use the Biological Processes subnetwork of the Gene Ontology. We allow relationships between biological functions of the following types: "regulates", "positively regulates", "negatively regulates", "part of", and "is a". The resulting biological function network consists of 29,189 nodes and 70,643 edges.

Protein–Protein Interactions Network We utilize the protein-protein network of 387,626 physical interactions between 17,660 proteins which is a part of the multiscale-interactome (Ruiz et al., 2021) generated by compiling seven major databases. The network contains interactions of human proteins with direct experimental evidence. It is additionally constrained to allow only physical interactions between proteins and filters out genetic and indirect interactions between proteins such as those identified via synthetic lethality experiments.

Population-Population Network We construct a population network based on the UMLS relations of terms extracted from the eligibility criteria and mapped to the UMLS concepts. We consider a subgraph of the UMLS network induced by the mapped eligibility criteria concepts and their child-parent relationships in the UMLS. The resulting population-population network has 31,925 nodes and 83,422 edges relating population concepts to each other based on the "is a" relationship. Additionally, we connect the population network to the rest of the biological networks, including MeSH, DrugBank, and Gene Ontology. For example, if the UMLS concept mentioned in the eligibility criteria defines a disease, then this node is connected to other diseases based on the disease-disease network extracted from the MeSH. The same holds for drugs and gene functions.

Protein-Function Network We construct a protein-function network containing 39,578 edges by associating proteins to the biological functions they affect by using the human version of the protein Gene Ontology Annotation Database (Huntley et al., 2015). We only allow experimentally

[3] http://geneontology.org/docs/ontology-documentation/

verified associations between genes and biological functions according to the following IDs: EXP (inferred from experiment), IDA (inferred from direct assay), IMP (inferred from mutant phenotype), IGI (inferred from genetic interaction), HTP (high throughput experiment), HDA (high throughput direct assay), HMP (high throughput mutant phenotype), and HGI (high throughput genetic interaction). We exclude any protein–biological function relationships inferred from physical interactions to avoid redundancy with the physical network of interacting proteins.

Drug-Function Network We construct a drug-function network containing 26,225 edges by associating drugs to phenotypes (biological functions) using the Chemical-Induced Phenotypes database provided by CTD (Davis et al., 2018). The chemical-phenotype database describes how chemicals can affect molecular, cellular, and physiological phenotypes curated from over 19,000 scientific articles with phenotypes represented using Gene Ontology terms. We obtain DrugBank Ids for the chemicals in the database using the mappings provided by CTD and restrict to relations between drugs and biological function present in our drug and biological functions hierarchies

Drug-Protein Network We generate a drug-protein interaction network containing 21,478 edges by associating drugs to their protein targets and associated enzymes using DrugBank (Wishart et al., 2018). There are different types of relations between drugs and proteins in DrugBank (for example, inhibitor, oxidizer, disruptor). However, we allow only 5 most frequent relation types (namely, inhibitor, substrate, antagonist, agonist, inducer) in our network and map all other relation types to a new catch-all relation type *other*. We map the UniProt Protein IDs in DrugBank to Entrez ID using the mapping provided by UniProt (Consortium, 2021b). We also filter drug-protein relationships to only include proteins that are represented in the network of physical interactions between proteins.

Disease-Protein Network We construct a disease-gene network containing 31,826 edges by associating diseases to genes they affect through effects like genomic alterations, altered expression, or post-translational modification by using DisGeNet (Piñero et al., 2019). To ensure high-quality disease-gene associations, we only consider the curated set of disease-gene associations provided by DisGeNet, which draws from expert-curated repositories: UniProt, the Comparative Toxicogenomics Database (human subset), Orphanet, the Clinical Genome Resource (ClinGen), Genomics England, the Cancer Genome Interpreter (CGI), and the Psychiatric Disorders Gene Association Network (PsyGeNET). We filter disease–gene relationships to only consider genes whose protein products were present in the network of physical interactions between proteins. We further filter the relationships to consider only the diseases for which a mapping to a corresponding disease in our disease hierarchy is present in the disease mappings provided by DisGeNet.

6.5.3 Answering queries using PlaNet knowledge graph

PlaNet knowledge graph can be used to answer diverse queries about the information stored in graph. For example, PlaNet can be used to easily retrieve all clinical trials in which a particular drug of interest caused serious adverse events (Supplementary Fig 1a). This can be useful to patients to investigate safety reports of a drug, or to easily identify trials that potentially under-reported results in the peer-reviewed publications which is a well investigated problem (Tang et al., 2015; Hartung et al., 2014). PlaNet connects drugs and diseases with proteins, so it can be used to investigate potential candidates for drug repurposing. For instance, raloxifene drug was originally developed for osteoporosis but then successfully repurposed for breast cancer (Pushpakom et al., 2019). Information that raloxifene targets *CYP19A1* protein, which is a prognostic marker in ER-positive breast cancer (Friesenhengst et al., 2018) is captured in the PlaNet (Supplementary Fig 1b). PlaNet also reveals other raloxifene trials that tested for diseases associated with *CYP19A1* protein and can be used to suggest new disease candidates associated with the same protein or even combinations of proteins that have not been investigated yet.

6.5.4 Baseline methods

We compare our model to drug-disease-outcome (DDO) and PubMedBERT baselines (Gu et al., 2021). In DDO baseline, we represent a trial arm by the one-hot encoding of the drugs, diseases and the outcomes associated with the arm. We then train a Random Forest classifier (Breiman, 2001) with 100 estimators and a max depth of 5.

In the PubmedBERT baseline, we use pretrained transformer language models fine-tuned on the clinical trials protocol text. Specifically, we use trial arm text which we obtain by concatenating intervention name, disease name, outcome measure, brief summary of the trial, arm name, arm intervention description, and eligibility criteria. We fine-tune them using the same task classifier architecture as in our model, *i.e.*, a shared hidden layer followed by a task-specific classifier. The model is fine-tuned for 50 epochs using the mean AUPRC score on the validation set as the early stopping criteria. We train with the batch size of 32 and restrict the maximum gradient norm to 1 and use a weight decay of 10^{-6}.

6.5.5 Hyperparameters

For the link prediction pretext task, we use a adversarial temperature α of 1 sampling 256 negative triplets for each positive triplet. We pretrain the encoder for $20,000$ steps with a batch size of $8,192$ using the Mean Reciprocal Ratio (MRR) (Radev et al., 2002) on the validation set as the early stopping criteria. We use an initial learning rate of 0.005 halving the learning rate at $3,000$, $6,000$, $12,000$ steps. We also restrict the maximum gradient norm to be 1.

For the outcome prediction task, we use a fully connected hidden layer with 800 dimensions with

LayerNorm and ReLU activation. We use a dropout of 0.5 in the outcome classifiers. We use a batch size of $1,024$ and train for 50 epochs using the mean AURPC score on a validation set as the stopping criteria. We use an initial learning rate of 0.001 halving the learning rate every 50-th epoch. We also restrict the maximum gradient norm to be 1 for classifier head as well. The learning rate of the encoder during fine-tuning is $1/10^{th}$ of the final learning rate of the encoder model during pretraining.

6.6 Our model generated clinical trials for investigating repurposing candidates

While PlaNet is able to predict drug effectiveness, we also investigated whether we can use PlaNet to search for drugs that have a potential to be more effective than an FDA approved drug and generate drug candidates for a particular diseases. We focused our question on capecitabine, an FDA approved treatment for metastatic breast cancer (Ershler, 2006). We created artificial clinical trials that have the same population properties as a capecitabine trial, but are testing a different drug. We asked PlaNet to rank drugs based on probability to be more effective than capecitabine. As candidate drugs we considered drugs that are within 2-hop neighbors of the breast cancer but have never appeared in the labeled efficacy prediction dataset with breast cancer, meaning that the model has never seen an outcome of the drug when applied to patients suffering from breast cancer. Among top ranked drugs PlaNet selected temozolomide, olaparib, ipilimumab, radium chloride Ra-223, enzalutamide, veliparib and avelumab. Temozolomide is an FDA approved drug used to treat brain cancers which has been investigated for its activity in metastatic breast cancers with still ongoing clinical trials (Garza-Morales et al., 2018; Han et al., 2018; Trudeau et al., 2006). Olaparib is an FDA approved drug for refractory metastatic breast cancer with deleterious germline mutations in BRCA1/2 (Waks and Winer, 2019; Robson et al., 2017). Ipilimumab, an FDA approved drug for melanoma, is currently in investigation with nivolumab for patients with metastatic recurrent HER2 negative inflammatory breast cancer(Adams et al., 2022). Radium chloride Ra-223, an FDA approved for castration resistant prostate cancer with bone metastases showed promise in breast cancer patients with bone metastases (Takalkar et al., 2015) with ongoing clinical trials. Enzalutamide, an AR inhibitor that impairs nuclear localization of AR, was used to elucidate the role of AR in preclinical models of ER positive and negative breast cancer (Cochrane et al., 2014) and has demonstrated clinical activity in patients with advanced AR-positive triple-negative breast cancer with a number of ongoing clinical trials (Traina et al., 2018). Veliparib and avelumab are both being investigated for metastatic breast cancer with active clinical trials (Dirix et al., 2018; Rugo et al., 2016). These results support immediate practical applicability of PlaNet in providing insights in potentially effective treatments.

Node Type	Count
Disease	5,751
Drug	14,300
Drug Class	4,825
Population	30,913
Protein	17,660
Function	28,734
Outcome	3,048
Trial Arm	205,809
Total	330,915

(a) Number of nodes of different types in our knowledge graph. Note that the nodes in our graph can have multiple types, *e.g.*, a node can have types *Population*, *Disease*, and *Adverse Event*. The total in the table refers to the total number of nodes in the graph and hence is not equal to the sum of the nodes of each type.

Sub-Network	# of edges
Trial	
Trial Arm-Drug	874,881
Trial Arm-Disease	2,360,695
Trial Arm-Inc. Population	2,778,103
Trial Arm-Exc. Population	5,370,161
Trial Arm-Outcome	911,260
Disease-Disease	17,510
Drug-Drug	133,661
Function-Function	67,118
Protein-Protein	387,626
Population-Population	92,744
Protein-Function	39,605
Drug-Function	26,142
Drug-Protein	21,478
Disease-Protein	22,062
Total	13,928,443

(b) Number of relations in our knowledge graph

Table 6.1: Number of nodes and relations of each subnetwork in our knowledge graph, constructed by combining data from `https://clinicaltrials.gov/` and existing biomedical knowledge bases such as UMLS.

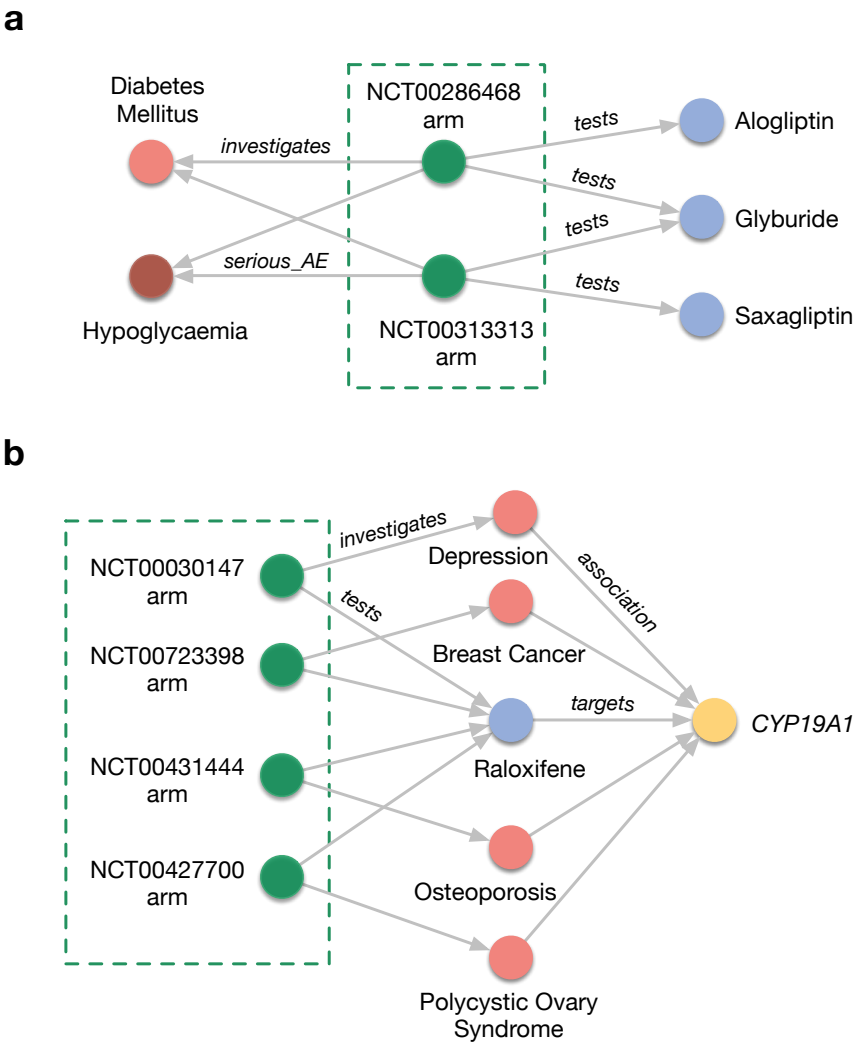


Figure 6.5: Examples of knowledge graph queries that PlaNet can answer. **(a)** PlaNet can be used to retrieve all clinical trials in which a drug of interest caused particular serious adverse event. The example shows trials in which glyburide drug caused serious hypoglycaemia. The publication associated with NCT00313313 trial reported no cases of serious hypoglycemia which is in disaccordance with the clinical trials database that reported 2 patients suffering from serious hypoglycemia (Hartung et al., 2014). **(b)** PlaNet can be used to investigate potential candidates for drug repurposing. In the example, raloxifene drug was originally developed for osteoporosis and repurposed for breast cancer (Pushpakom et al., 2019) which is captured in the PlaNet. Raloxifene targets *CYP19A1* protein, which is a prognostic marker in ER-positive breast cancer (Friesenhengst et al., 2018).

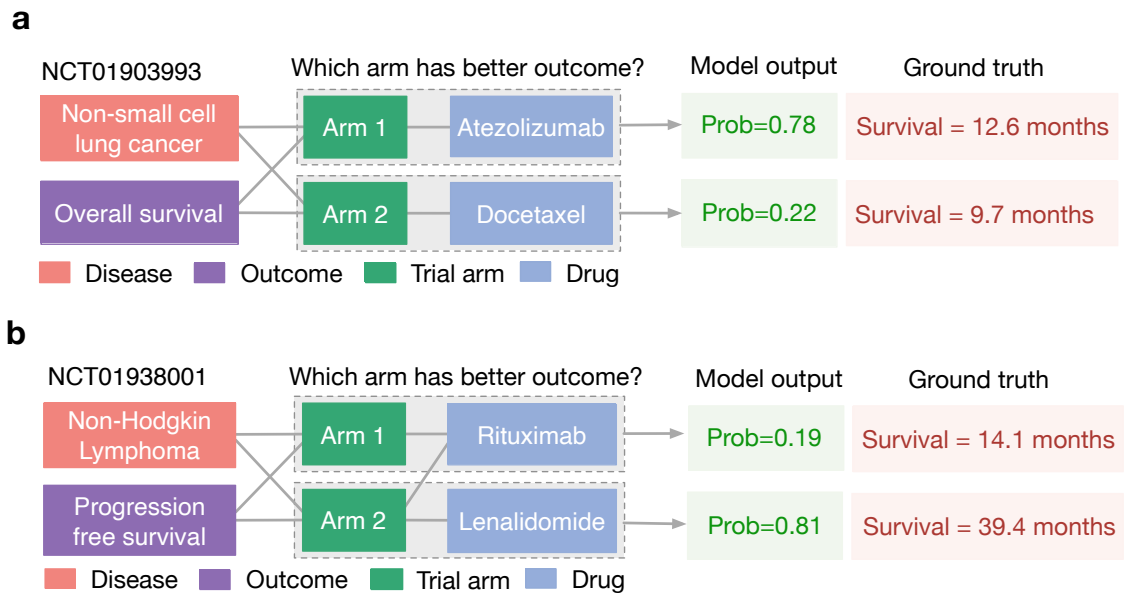


Figure 6.6: Examples on which PlaNet is the only model that correctly predicted outcome. **(a)** PlaNet correctly predicted higher overall survival of non-small cell lung cancer patients in atezolizumab arm compared to docetaxel arm. Model output corresponds to probabilities that a given arm has higher overall survival. **(b)** PlaNet correctly predicted higher progression free survival of non-Hodgkin lymphoma patients for the combination of rituximab and lenalidomide drugs compared to lenalidomide drug alone. Model output corresponds to probabilities that a given arm has higher progression free survival.

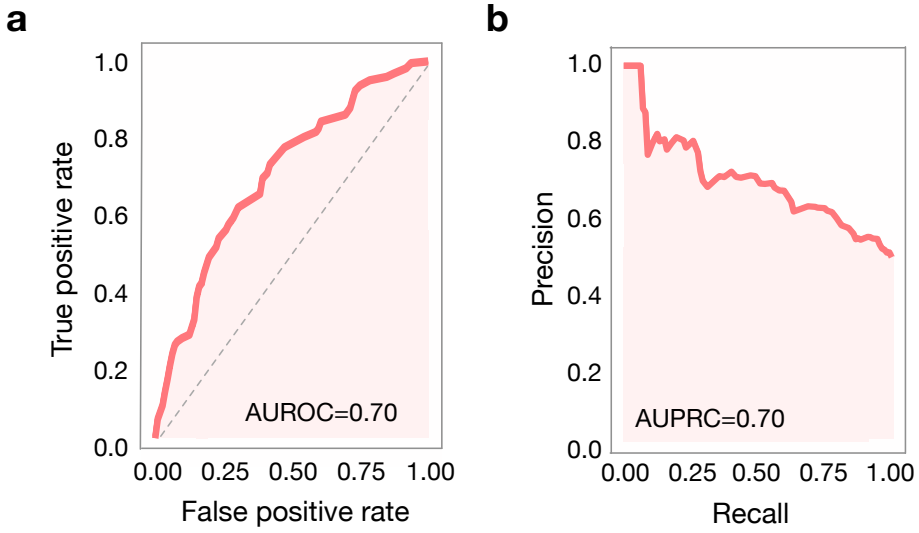


Figure 6.7: Performance of the PlaNet on the efficacy prediction task measured as **(a)** area under receiver operating characteristic curve (AUROC) and **(b)** area under precision-recall curve (AUPRC). Higher value indicates better performance, where 1 is perfect performance. For AUROC, 0.5 is random baseline. Efficacy task is defined as predicting which trial arm will have more beneficial survival outcome.

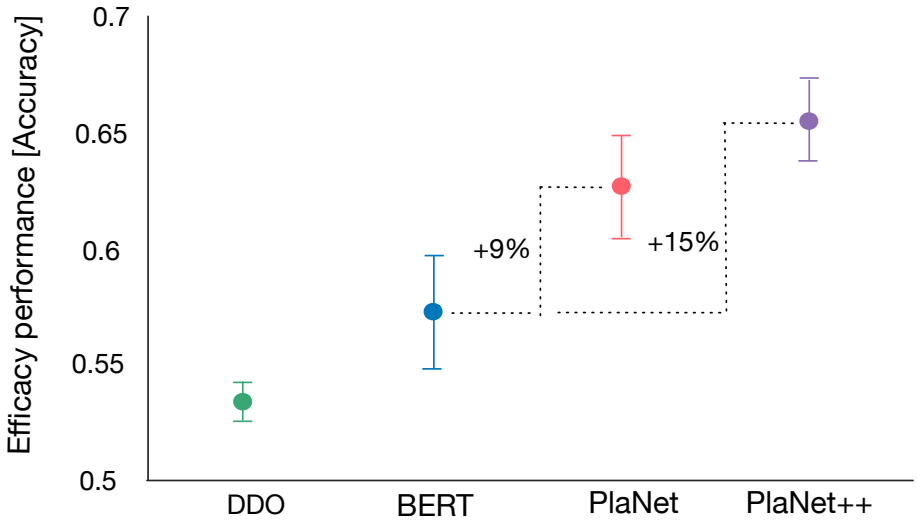


Figure 6.8: Performance comparison of the PlaNet with DDO and PubMedBERT baselines. Combined model is obtained by concatenating the PlaNet protocol embeddings with PubMedBERT embedding from text and fine-tuning them jointly. Performance is measured as the mean accuracy score across 10 runs of each model on different test data samples. Error bars are 95% bootstrap confidence intervals.

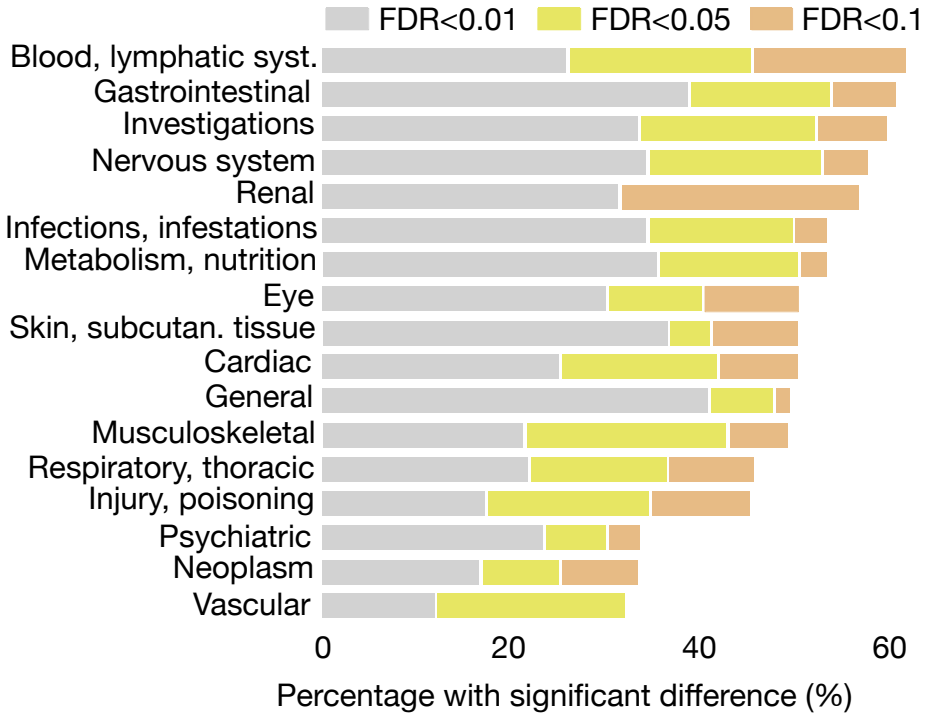


Figure 6.9: Comparison of the adverse events frequency distributions between trials that apply drug to populations suffering from the same disease and trials in which drug is applied to populations that suffer from a different disease while keeping the drug fixed in both cases. In such a way, we monitor whether there is a significant difference in adverse event frequency distributions when same drug is applied to different populations. The x axis denotes percentage of examples that have significant difference in frequency distribution, while y axis shows broad adverse events categories.

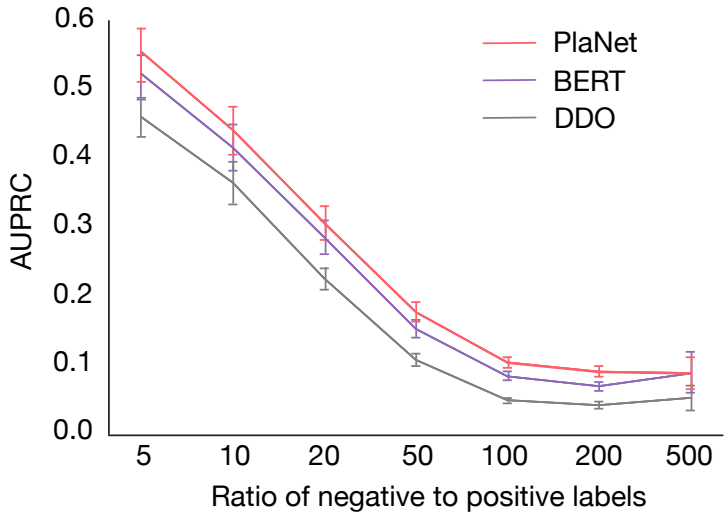


Figure 6.10: Performance of the PlaNet and baseline models on the adverse events prediction task as a function of the ratio of negative to positive labels. Performance is measured as the mean AUPRC score across all side effects with the given ratio. Error bars are 95% bootstrap confidence intervals.The AUPRC baseline equals the number of positive examples in the data.

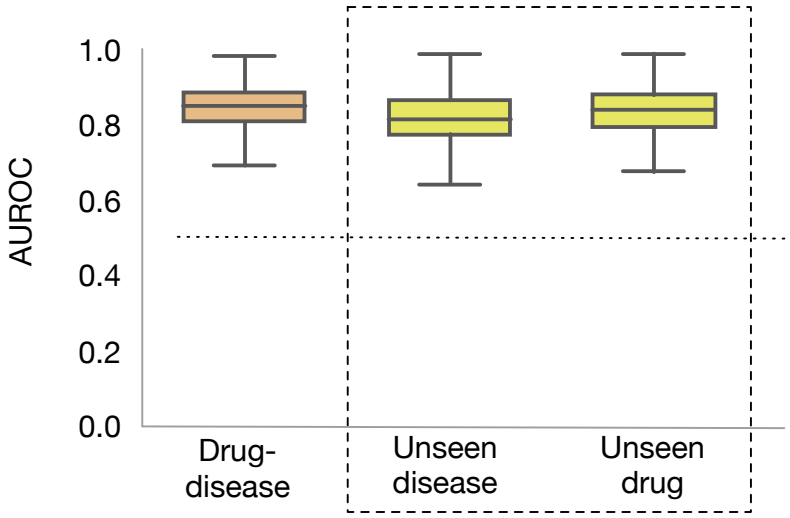


Figure 6.11: Comparison of different data splits on the PlaNet performance. Drug-disease split ensures unique drug-disease pairs in the test set compared to the train set, while unseen disease and drug splits require generalization to never-before-seen drugs and never-before-seen diseases, respectively. In all splits, there is no trial leakage between the train and test set, *i.e.,* all arms of the same trial are in the same split. The boxes show the quartiles of the performance distribution across different adverse events. Whiskers show the rest of the distribution.

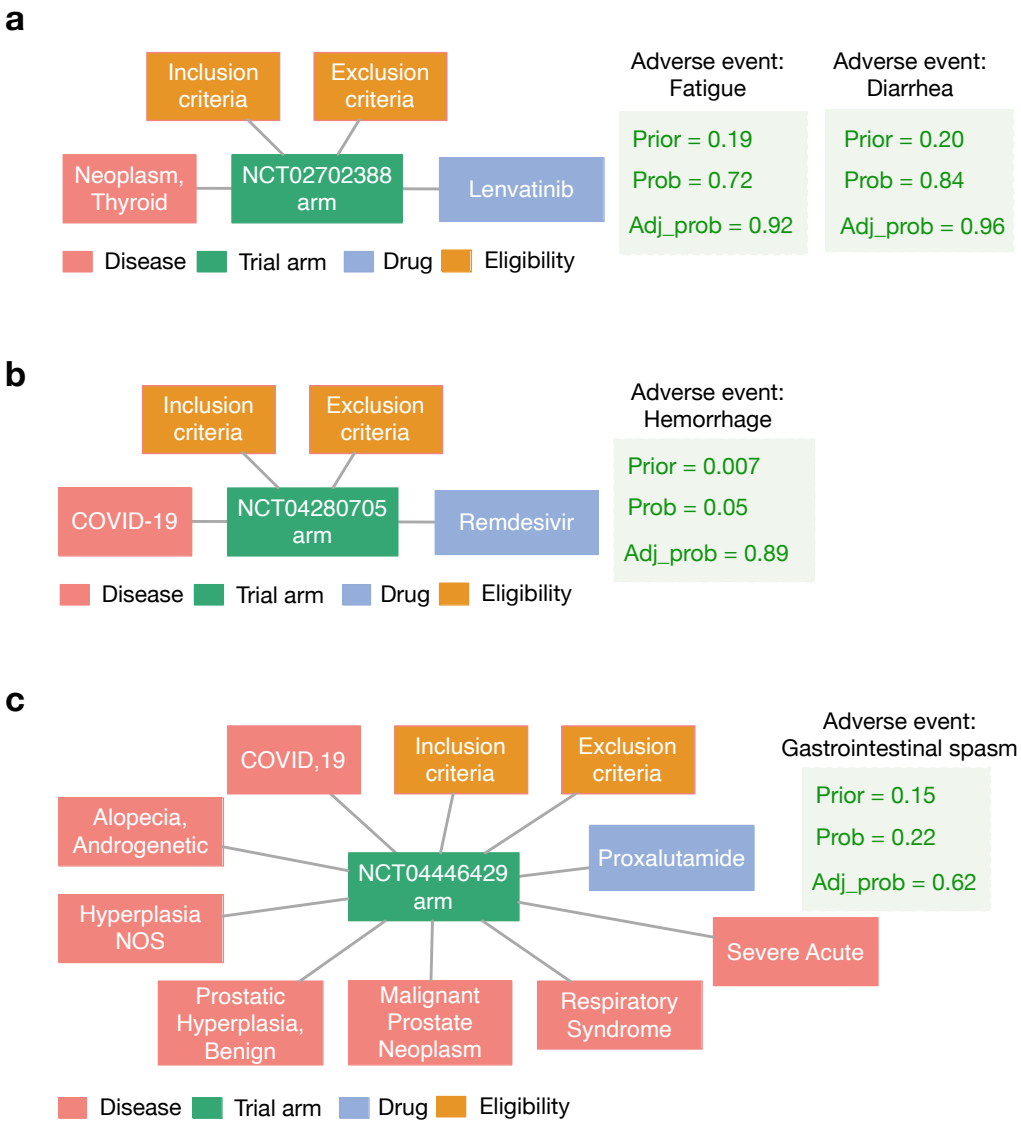


Figure 6.12: Examples of future trial predictions. Model outputs probabilities that an adverse event will be enriched in a given arm compared to no-treatment arm. Prior corresponds to estimated probability of an adverse event when no treatment is given to the population. Adjusted probabilities are probabilities adjusted from the the prior probability. Inclusion and exclusion terms are joined for the visualization purposes. **(a)** In a trial that tested safety of lenvatinib for thyroid cancer patients, PlaNet correctly predicted fatigue and diarrhea as side effects with a high confidence, which were actually reported in 58.3% and 36.1% patients (Giani et al., 2021), respectively. **(b, c)** In recent COVID-19 trials, PlaNet correctly increased the probability of **(b)** hemorrhage and **(c)** gastrointestinal spasm. The model has never seen any COVID-19 example during training.

Chapter 7

Multimodal medical question answering

In this chapter, we present the application of textual and visual knowledge fusion (§5) for building multimodal medical question answering systems.

7.1 Introduction

Large language models have demonstrated capabilities in solving an abundance of tasks, including question answering, by being provided only a few demonstration examples as context (Bommasani et al., 2021). This is known as in-context learning (Brown et al., 2020), through which a model learns to perform a task from demonstrations during prompting and without tuning the model parameters. In the medical domain, this bears great potential to create practically useful question answering systems: it will enable medical AI models to handle the various rare cases faced by clinicians every day in a unified way, to provide relevant rationales to justify their statements, and to easily customize model generations to specific use cases (Moor et al., 2023a).

Developing models that can answer diverse questions in a medical setting is, however, challenging due to the inherent multimodality of medical data. For example, medical data has various data types (text, image, video, database, molecule), scales (molecule, gene, cell, tissue, patient, population) (Kong et al., 2011; Ruiz et al., 2021), and styles (professional and lay language) (Lavertu and Altman, 2019; Li et al., 2019b).

Previous efforts to create large multimodal medical models, such as ChexZero (Tiu et al., 2022) and BiomedCLIP (Zhang et al., 2023a), have made strides in their respective domains. ChexZero specializes in chest X-ray interpretation, while BiomedCLIP has been trained on more diverse images paired with captions from the biomedical literature. Other models have also been developed for

electronic health record (EHR) data (Steinberg et al., 2021) and surgical videos (Kiyasseh et al., 2023). However, none of these models have embraced in-context learning for the multimodal medical domain. Existing medical VLMs, such as MedVINT (Zhang et al., 2023b), are typically trained on paired image-text data with a single image in the context, as opposed to more general streams of text that are interleaved with multiple images. Therefore, these models were not designed and tested to perform multimodal in-context learning with few-shot examples[1].

Here, we propose Med-Flamingo, the first medical language model that can perform multimodal in-context learning to perform various tasks including question answering. Med-Flamingo is a vision-language model based on Flamingo (Alayrac et al., 2022b) that can naturally ingest data with interleaved modalities (images and text), to generate text conditioned on this multimodal input. Building on the success of Flamingo, which was among the first vision-language models to exhibit in-context learning and few-shot learning abilities, Med-Flamingo extends these capabilities to the medical domain by pre-training on multimodal knowledge sources across medical disciplines.

In preparation for the training of Med-Flamingo, our initial step involved constructing a unique, interleaved image-text dataset, which was derived from an extensive collection of over $4K$ medical textbooks (Section 7.3). Given the critical nature of accuracy and precision within the medical field, it is important to note that the quality, reliability, and source of the training data can considerably shape the results. Therefore, to ensure accuracy in medical facts, we meticulously curated our dataset from respected and authoritative sources of medical knowledge, as opposed to relying on potentially unreliable web-sourced data.

In our experiments, we evaluate Med-Flamingo on generative medical visual question-answering (VQA) tasks by directly generating open-ended answers, as opposed to scoring artificial answer options *ex post*–as CLIP-based medical vision-language models do. We design a new realistic evaluation protocol to measure the model generations' clinical usefulness. For this, we conduct an in-depth human evaluation study with clinical experts which results in a human evaluation score that serves as our main metric. In addition, due to existing medical VQA datasets being narrowly focused on image interpretation among the specialties of radiology and pathology, we create Visual USMLE[2], a challenging generative VQA dataset of complex USMLE-style problems across specialties, which are augmented with images, case vignettes, and potentially with lab results.

Averaged across three generative medical VQA datasets, few-shot prompted Med-Flamingo achieves the best average rank in clinical evaluation score (rank of 1.67, best prior model has 2.33), indicating that the model generates answers that are most preferred by clinicians, with up to 20% improvement over prior models. Furthermore, Med-Flamingo is capable of performing medical reasoning, such as answering complex medical questions (such as visually grounded USMLE-style questions) and providing explanations (i.e., rationales), a capability not previously demonstrated by other

[1]For example, a challenge with multimodal in-context learning for existing medical vision language models is the potential for image information to leak across examples, potentially misleading the model.

[2]USMLE stands for "United States Medical Licensing Examination".

multimodal medical pretrained models. However, it is important to note that Med-Flamingo's performance may be limited by the availability and diversity of training data, as well as the complexity of certain medical tasks. All investigated models and baselines would occasionally hallucinate or generate low-quality responses. Despite these limitations, our work represents a significant step forward in the development of multimodal medical language models and their ability to perform multimodal in-context learning in the medical domain. We release the Med-Flamingo-9B checkpoint for further research, and make our code available under `https://github.com/snap-stanford/med-flamingo`. In summary, our paper makes the following contributions:

1. We present the first multimodal few-shot learner adapted to the medical domain, which promises novel clinical applications such as rationale generation and conditioning on retrieved multimodal context.

2. We create a novel dataset that enables the pre-training of a multimodal few-shot learner for the general medical domain.

3. We create a novel USMLE-style evaluation dataset that combines medical VQA with complex, across-specialty medical reasoning.

4. We highlight shortcomings of existing evaluation strategies, and conduct an in-depth clinical evaluation study of open-ended VQA generations with medical raters using a dedicated evaluation app.

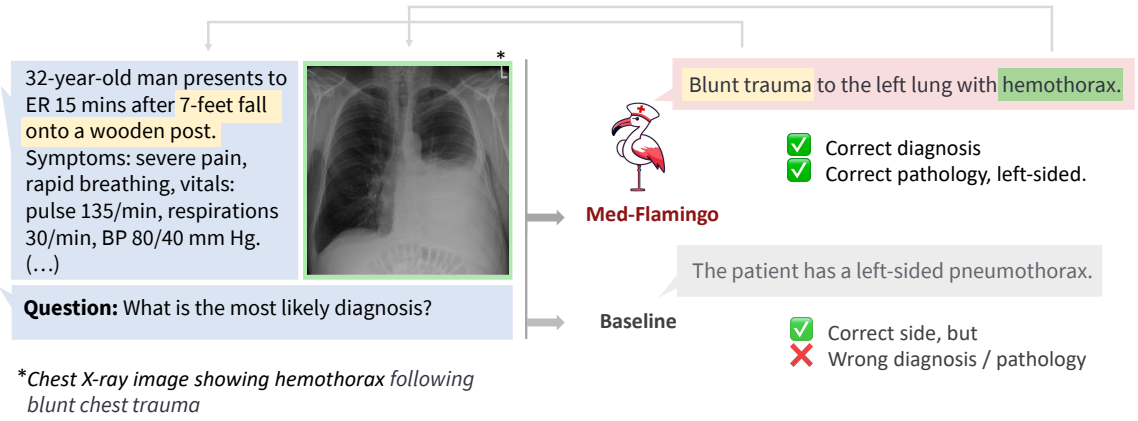


Figure 7.1: Example of how Med-Flamingo answers complex multimodal medical questions by generating open-ended responses conditioned on textual and visual information. The baseline response was given by the OpenFlamingo model, both models were few-shot prompted with 4 shots.

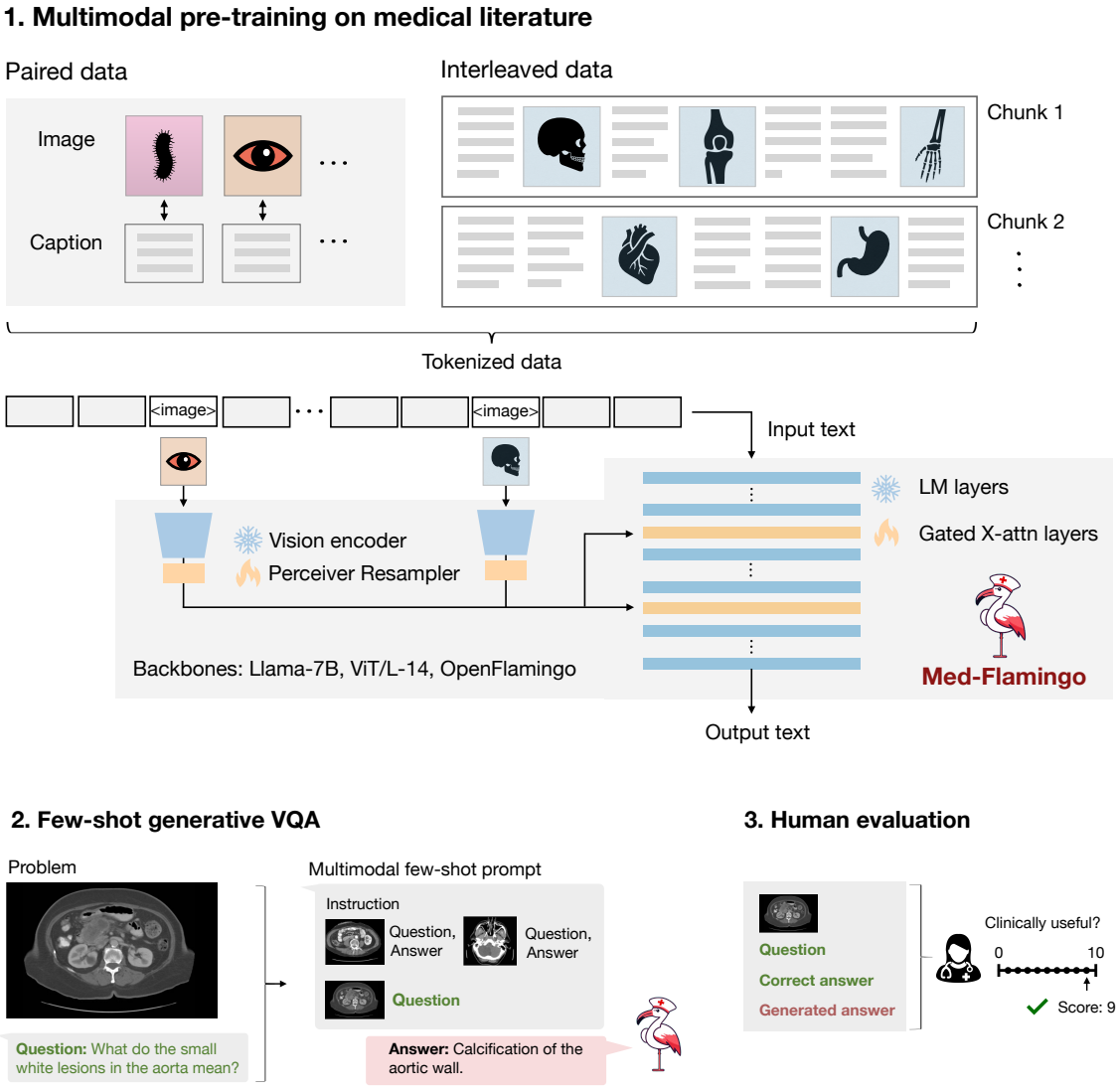


Figure 7.2: Overview of the Med-Flamingo model and the three steps of our study. First, we pre-train our Med-Flamingo model using paired and interleaved image-text data from the general medical domain (sourced from publications and textbooks). We initialize our model at the OpenFlamingo checkpoint continue pre-training on medical image-text data. Second, we perform few-shot generative visual question answering (VQA). For this, we leverage two existing medical VQA datasets, and a new one, Visual USMLE. Third, we conduct a human rater study with clinicians to rate generations in the context of a given image, question and correct answer. The human evaluation was conducted with a dedicated app and results in a clinical evaluation score that serves as our main metric for evaluation.

7.2 Related works

The success of large language models (LLMs) (Brown et al., 2020; Liang et al., 2022; Qin et al., 2023) has led to significant advancements in training specialized models for the medical domain. This has resulted in the emergence of various models, including BioBERT (Lee et al., 2020), ClinicalBERT (Huang et al., 2019a), PubMedBERT (Gu et al., 2021), BioLinkBERT (Yasunaga et al., 2022b), DRAGON (Yasunaga et al., 2022a), BioMedLM (Bolton et al., 2022), BioGPT (Luo et al., 2022), and Med-PaLM (Singhal et al., 2022). Although these medical language models are typically smaller than general-purpose LLMs like GPT-3 (Brown et al., 2020), they can match or even surpass their performance on medical tasks, such as medical question answering.

Recently, there has been a growing interest in extending language models to handle vision-language multimodal data and tasks (Su et al., 2019; Ramesh et al., 2021; Alayrac et al., 2022b; Aghajanyan et al., 2022; Yasunaga et al., 2023). Furthermore, many medical applications involve multimodal information, such as radiology tasks that require the analysis of both X-ray images and radiology reports (Tiu et al., 2022). Motivated by these factors, we present a medical vision-language model (VLM). Existing medical VLMs include BiomedCLIP (Zhang et al., 2023a), MedVINT (Zhang et al., 2023b). While BiomedCLIP is an encoder-only model, our focus lies in developing a generative VLM, demonstrating superior performance compared to MedVINT. Finally, Llava-Med is another recent medical generative VLM (Li et al., 2023), however the model was not yet available for benchmarking.

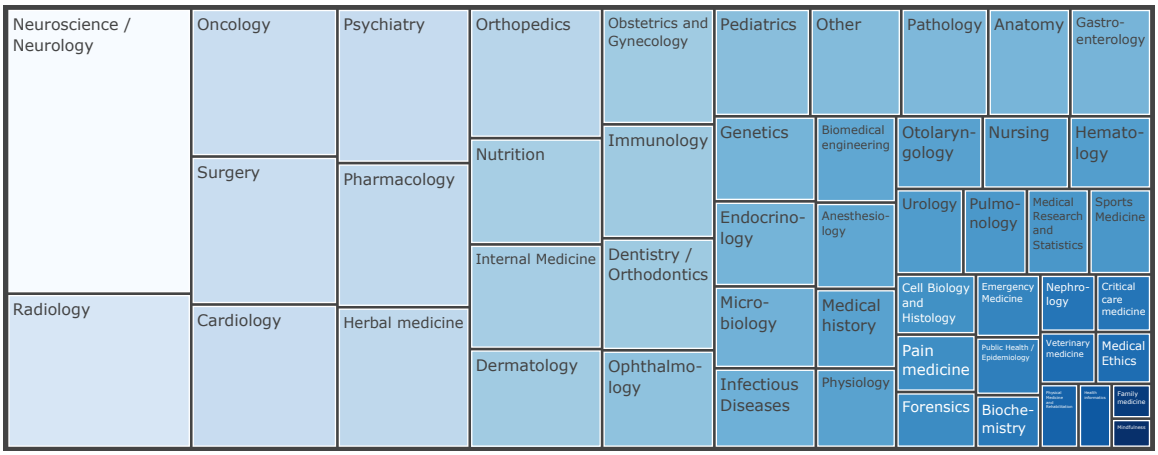


Figure 7.3: Overview of the distribution of medical textbook categories of the MTB dataset. We classify each book title into one of the 49 manually created categories or "other" using the Claude-1 model.

7.3 Approach

To train a Flamingo model adapted to the medical domain, we leverage the pre-trained OpenFlamingo-9B model checkpoint (Awadalla et al., 2023), which is a general-domain VLM that was built on top of the frozen language model LLaMA-7B (Touvron et al., 2023) and frozen vision encoder CLIP ViT/L-14 (Radford et al., 2021a). We perform continued pre-training in the medical domain which results in the model we refer to as Med-Flamingo.

7.3.1 Data

We pre-train Med-Flamingo by jointly training on interleaved image-text data and paired image-text data. As for the interleaved dataset, we created a interleaved dataset from a set of medical textbooks, which we subsequently refer to as MTB. As for the paired datasets, we used PMC-OA (Lin et al., 2023).

MTB We construct a new multimodal dataset from a set of 4 721 textbooks from different medical specialties (see Figure 7.3). During preprocessing, each book is first converted from PDF to HTML with all tags removed, except the image tags are converted to <image> tokens. We then carry out data cleaning via deduplication and content filtering. Finally, each book with cleaned text and images is then chopped into segments for pretraining so that each segment contains at least one image and up to 10 images and a maximum length. In total, MTB consists of approximately 0.8M images and 584M tokens. We use 95% of the data for training and 5% of the data for evaluation during the pre-training.

PMC-OA We adopt the PMC-OA dataset (Lin et al., 2023) which is a biomedical dataset with 1.6M image-caption pairs collected from PubMedCentral's OpenAccess subset. We use 1.3M image-caption pairs for training and 0.16M pairs for evaluation following the public split[3].

7.3.2 Objectives

We follow the original Flamingo model approach (Alayrac et al., 2022a), which considers the following language modelling problem:

$$p\left(y_\ell \mid x_{<\ell}, y_{<\ell}\right) = \prod_{\ell=1}^{L} p\left(y_\ell \mid y_{<\ell}, x_{<\ell}\right),$$

where y_ℓ refers to the ℓ-th language token, $y_{<\ell}$ to the set of preceding language tokens, and $x_{<\ell}$ to the set of preceding visual tokens. As we focus on modelling the medical literature, here we consider only image-text data (i.e., no videos).

[3] https://huggingface.co/datasets/axiong/pmc_oa_beta

Following Alayrac et al. (2022a), we minimize a joint objective $\mathcal{L}$ over paired and interleaved data:

$$\mathcal{L} = \mathbb{E}_{(x,y)\sim D_p}[S(x,y)] + \lambda \cdot \mathbb{E}_{(x,y)\sim D_i}[S(x,y)],$$

where $S(x,y) = -\sum_{\ell=1}^{L} \log p\left(y_\ell \mid y_{<\ell}, x_{<\ell}\right)$, and D_p and D_i stand for the paired and interleaved dataset, respectively. In our case, we use $\lambda = 1$.

7.3.3 Training

We performed multi-gpu training on a single node with 8x 80GB NVIDIA A100 GPUs. We trained the model using DeepSpeed ZeRO Stage 2: Optimizer states and gradients are sharded across devices. To further reduce memory load, we employed the 8-bit AdamW optimizer as well as the memory-efficient attention implementation of PyTorch 2.0. Med-Flamingo was initialized at the checkpoint of the Open-Flamingo model and then pre-trained for 2700 steps (or 6.75 days in wall time, including the validation steps), using 50 gradient accumulation steps and a per-device batch size of 1, resulting in a total batch size of 400. The model has $1.3B$ trainable parameters (gated cross attention layers and perceiver layers) and roughly $7B$ frozen parameters (decoder layers and vision encoder), which results in a total of $8.3B$ parameters. Note that this is the same number parameters as in the OpenFlamingo-9B model (version 1).

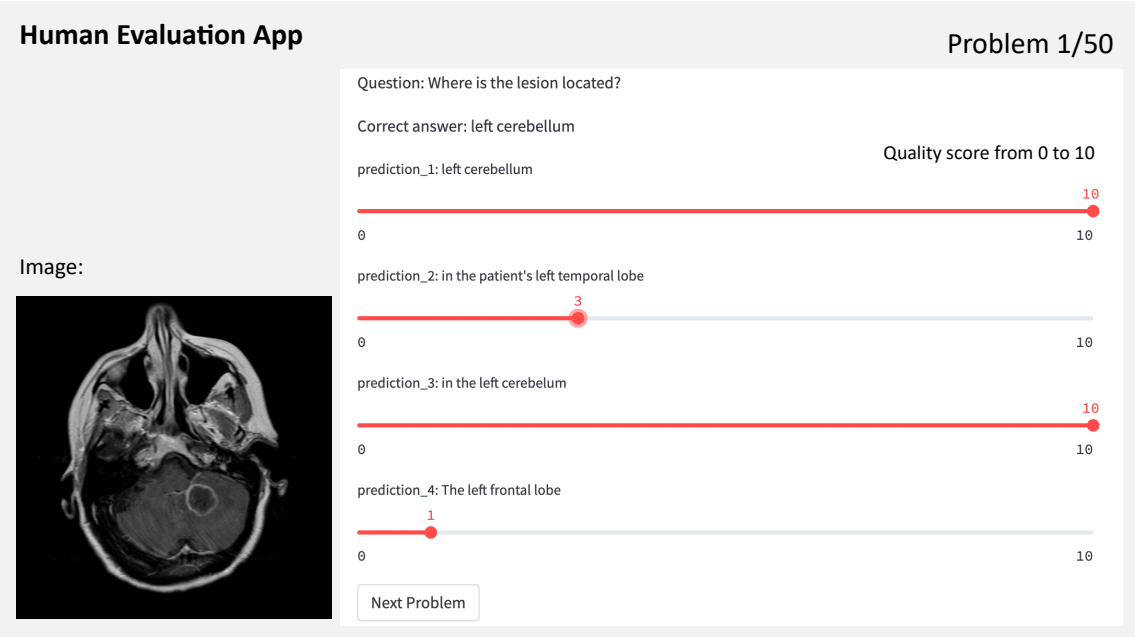


Figure 7.4: Illustration of our Human evaluation app that we created for clinical experts to evaluate generated answers.

7.4 Evaluation

7.4.1 Automatic Evaluation

Baselines To compare generative VQA abilities against the literature, we consider different variants of the following baselines:

1. MedVINT (Zhang et al., 2023b), a visual instruction-tuned VLM based on Llama. As this model was not designed to do few-shot learning (e.g. the image information is prepended to the overall input), we report two modes for MedVINT: zero-shot and fine-tuned, where the model was fine-tuned on the training split of the VQA dataset. Since the rather small Visual-USMLE dataset has no separate training split, we ommit the fine-tuned baseline for that dataset. We used the MedVInT-TD model with PMC-LLaMA and PMC-CLIP backbones.

2. OpenFlamingo (Awadalla et al., 2023), a powerful VLM which was trained on general-domain data, and which served as the base model to train Med-Flamingo. We report both zero-shot and few-shot performance. We expect Flamingo-type models to shine in the few-shot setting which they are designed for (as already the pre-training task includes multiple interleaved image-text examples).

Evaluation datasets To evaluate our model and compare it against the baselines, we leverage two existing VQA datasets from the medical domain (VQA-RAD and PathVQA). Upon closer inspection of the VQA-RAD dataset, we identified severe data leakage in the official train / test splits, which is problematic given that many recent VLMs fine-tune on the train split. To address this, we created a custom train / test split by seperately splitting images and questions (each 90% / 10%) to ensure that no image or question of the train split leaks into the test split. On these datasets, 6 shots were used for few-shot, whereas as the in-context examples were randomly drawn from the respective train splits.

Furthermore, we create Visual USMLE, a challenging multimodal problem set of 618 USMLE-style questions which are not only augmented with images but also with a case vignette and potentially tables of laboratory measurements. The Visual USMLE dataset was created by adapting problems from the Amboss platform (using licenced user access). To make the Visual USMLE problems more actionable and useful, we rephrased the problems to be open-ended instead of multiple-choice. This makes the benchmark harder and more realistic, as the models have to come up with differential diagnoses and potential procedures completely on their own—as opposed to selecting the most reasonable answer choice from few choices. Figure 7.7 gives an overview of the broad range of specialties that are covered in the dataset, greatly extending existing medical VQA datasets which are narrowly focused on radiology and pathology. For this comparatively small dataset, instead of creating a training split for finetuning, we created a small train split of 10 problems which can be

used for few-shot prompting. For this dataset (with considerably longer problems and answers), we used only 4 shots to fit in the context window.

Dataset	Model	Clinical eval. score	BERT-sim	Exact-match
VQA-RAD	MedVINT zero-shot	4.63	0.628	0.167
	MedVINT fine-tuned ($\sim 2K$ samples)	2.87	0.611	0.133
	OpenFlamingo zero-shot	4.39	0.490	0.000
	OpenFlamingo few-shot	<u>4.69</u>	<u>0.645</u>	**0.200**
	Med-Flamingo zero-shot	3.82	0.480	0.000
	Med-Flamingo few-shot	**5.61**	**0.650**	**0.200**
Path-VQA	MedVINT zero-shot	0.13	0.608	0.272
	MedVINT fine-tuned ($\sim 20K$ samples)	1.23	**0.723**	**0.385**
	OpenFlamingo zero-shot	**2.16**	0.474	0.009
	OpenFlamingo few-shot	<u>2.08</u>	0.669	0.288
	Med-Flamingo zero-shot	1.72	0.521	0.120
	Med-Flamingo few-shot	1.81	<u>0.678</u>	<u>0.303</u>
Visual USMLE	MedVINT zero-shot	0.41	0.421	-
	OpenFlamingo zero-shot	<u>4.31</u>	**0.512**	-
	OpenFlamingo few-shot	3.39	0.470	-
	Med-Flamingo zero-shot	4.18	<u>0.473</u>	-
	Med-Flamingo few-shot	**4.33**	0.431	-

Table 7.1: Performance metrics across VQA-Rad, PathVQA, and Visual USMLE datasets. Best scores are highlighted in bold. Emphasis is placed on the clinical evaluation score. BERT-sim likely does not capture all fine-grained medical details. Exact-match is brittle, though provides a conservative measure. Exact-match was uninformative (constant 0) for Visual USMLE due to long correct answers. The fine-tuned baseline did not surpass zero-shot performance in VQA-Rad, possibly due to its small size and custom splits to prevent leakage. Notably, the PathVQA dataset revealed a pronounced performance deficit in pathology, underscoring that prior classification metrics might have overestimated VLMs' efficacy in this domain.

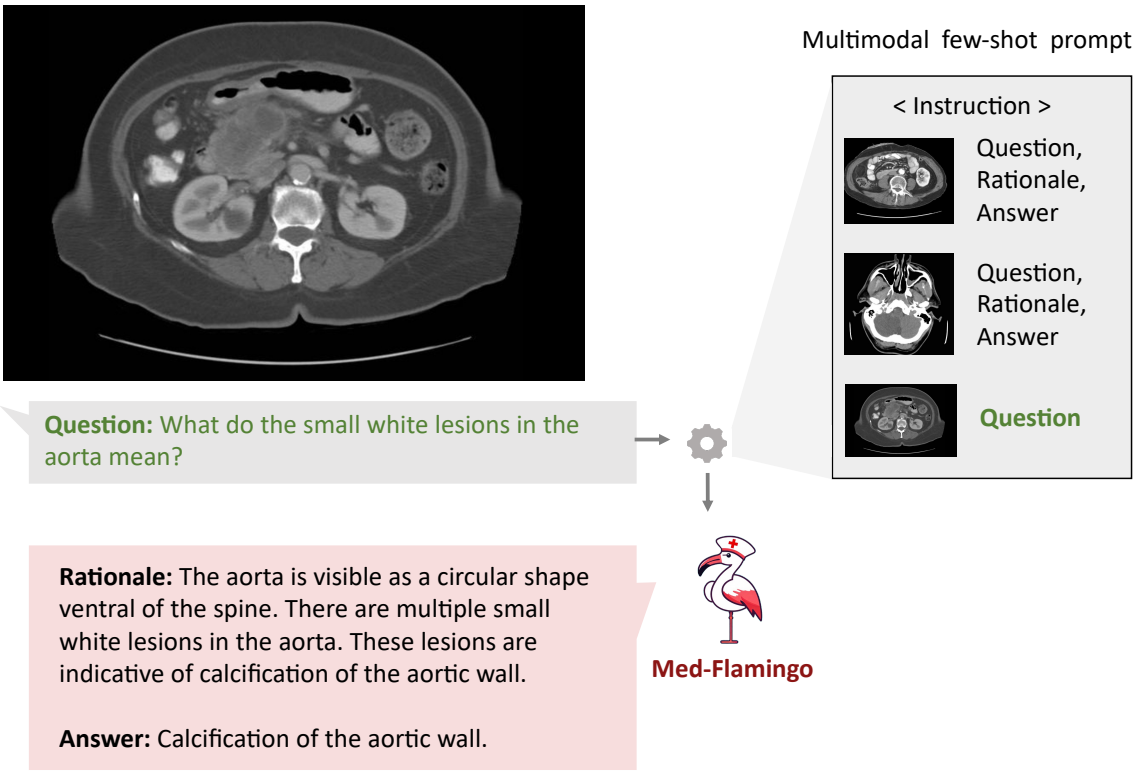


Figure 7.5: Multimodal medical few-shot prompting illustrated with an example. Few-shot prompting here allows users to customize the response format, *e.g.*, to provide rationales for the provided answers. In addition, multimodal few-shot prompts potentially offer the ability to include relevant context retrieved from the medical literature.

Evaluation metrics Previous works in medical vision-language modelling typically focused scoring all available answers of a VQA dataset to arrive at a classification accuracy. However, since we are interested in *generative* VQA (as opposed to post-hoc scoring different potential answers), for sake of clinical utility, we employ the following evaluation metrics that directly assess the quality of the generated answer:

1. Clinical evaluation score, as rated by three medical doctors (including one board-certified radiologist) using a human evaluation app that we developed for this study. More details are provided in Section 7.4.2.

2. BERT similarity score (BERT-sim), the F1 BERT score between the generated answer and the correct answer (Zhang et al., 2020).

3. Exact-match, the fraction of generated answers that exactly match (modulo punctuation) the correct answer. This metric is rather noisy and conservative as useful answers may not lexically match the correct answer.

7.4.2 Human evaluation

We implemented a human evaluation app using Streamlit to visually display the generative VQA problems for clinical experts to rate the quality of the generated answers with scores from 0 to 10. Figure 7.4 shows an examplary view of the app. For each VQA problem, human raters are provided with the image, the question, the correct answer, and a set of blinded generations (e.g., appearing as "prediction_1" in Figure 7.4), that appear in randomized order. As for human raters, we employed three medical doctors that were affiliated with the same academic center, however received their medical training in different countries. The team of raters included one board-certified radiologist.

7.4.3 Deduplication and leakage

During the evaluation of the Med-Flamingo model, we were concerned that there may be leakage between the pre-training datasets (PMC-OA and MTB) and the down-stream VQA datasets used for evaluation; this could inflate judgements of model quality, as the model could memorize image-question-answer triples.

To alleviate this concern, we performed data deduplication based upon pairwise similarity between images from our pre-training datasets and the images from our evaluation benchmarks. To detect similar images, in spite of perturbations due to cropping, color shifts, size, etc, we embedded the images using Google's Vision Transformer, preserving the last hidden state as the resultant embedding (Dosovitskiy et al., 2021). We then found the k-nearest neighbors to each evaluation image from amongst the pre-training images (using the FAISS library) (Johnson et al., 2019). We then

sorted and visualized image-image pairs by least euclidean distance; we found that images might be duplicates until a pairwise distance value of 80; beyond this point, there were no duplicates.

This process revealed that the pretraining datasets leaked into the PVQA evaluation benchmark. Out of 6700 total images in PVQA test set, we judged 194 to be highly similar to images in the pretraining datasets, and thus, we removed them from our down-stream evaluation.

7.5 Results

In our experiments, we focus on generative medical visual question answering (VQA). While recent medical VLMs predominantly performed VQA in a non-generative but rather discriminative manner (i.e., by scoring different answer choices), we believe that this ex-post classification to carry less clinical usefulness, than directly generating responses. On the other hand, generative VQA is more challenging to evaluate, as automated metrics suffer from significant limitations as they do not fully capture the domain-specific context. Thus, we perform a human evaluation study where clinical experts review model generations (blinded) and score them (between 0 and 10) in terms of clinical usefulness.

Conventional VQA datasets Table 7.1 shows the results for VQA-RAD, the radiological VQA dataset for which we created custom splits to address leakage (see Section 7.4). Med-Flamingo few-shot shows strong results, improving the clinical eval score by $\sim 20\%$ over the best baseline. In this dataset, the auxiliary metrics are rather aligned with clinical preference. Finetuning the MedVINT baseline did not lead to improved performance on this dataset which may be due to its small size. MedVINT zero-shot outperforms the other zero-shot ablations which may be partially attributed to its instruction tuning step on PMC-VQA.

Table 7.1 shows for the results for Path-VQA, the pathology VQA dataset. Compared to the other datasets, all models overall perform poorer on the Path-VQA dataset in terms of clinical evaluation score. We hypothesize that this has to do with the fact the models are not pre-trained on actual large-scale and fine-grained pathology image datasets, but only on a rather small amount of pathology literature (which may not be enough to achieve strong performance). For instance, Figure 7.3 shows that only a small fraction of our training data covers pathology. In the automated metrics (BERT-sim and exact-match), Med-Flamingo improves upon the OpenFlamingo baseline, however the overall quality does not improve (as seen in the clinical evaluation score). MedVINT was fine-tuned on a sizeable training split which results in strong automated metrics, but did not result in a clinical evaluation score that matches any Flamingo variant.

Visual USMLE Table 7.1 shows the results for the Visual USMLE dataset. Med-Flamingo (few-shot) results in the clinically most preferrable generations, whereas OpenFlamingo (zero-shot) is a close runner-up. As the ground truth answers were rather lengthy paragraphs, exact match was

not an informative metric (constant 0 for all methods). The few-shot prompted models lead to lower automated scores than their zero-shot counterparts, which we hypothesize has to do with the fact that the USMLE problems are long (long vignettes as well as long answers) which forced us to summarize the questions and answers when designing few-shot prompts (for which we used GPT-4). Hence, it's possible that those prompts lead to short answers that in terms of BERT-sim score may differ more from the correct answer than a more wordy zero-shot generation.

Across datasets Overall, we find that Med-Flamingo's multimodal in-domain few-shot learning abilities lead to favorable generative VQA performance, leading to the lowest average rank of 1.67 in terms of clinical evaluation score as averaged across all evaluation datasets. As runner-up, OpenFlamingo zero-shot achieves a rank of 2.33.

Qualitative analysis Finally, we showcase few examples of Med-Flamingo generations in more detail in Figures 7.1, 7.5, and 7.8. Figure 7.5 exemplifies that a medical few-shot learner like Med-Flamingo can be prompted to generate rationale for its VQA answer. The shown example is impressive in that the rationale is visually guiding the reader towards the object of interest (calcification of the aortic wall). We note, however, that at this stage, few-shot multimodal prompted rationales may not be robust, especially when a model arrives at a wrong answer.

Figures 7.1 and 7.8 showcase two example problems from the Visual USMLE dataset. The problem descriptions were slightly rephrased and summarized using GPT-4 for display. In Figure 7.8, Med-Flamingo generates the correct answer while not mentioning the underlying diagnosis (urothelial cancer) as it was not asked for. By contrast, we observed baselines to directly diagnose the patient (instead of answering the actual question in a targeted way). The problem in Figure 7.1 illustrates that Med-Flamingo has the ability to integrate complex medical history information together with visual information to synthesize a comprehensive diagnosis that draws from the information of both modalities. As for failure modes, we occasionally observed that information from the in-context examples can leak into the final generation.

7.6 Discussion

In this paper, we presented Med-Flamingo, the first medically adapted multimodal few-shot learner. While this is an early proof-of-concept for a medical multimodal few-shot learner, we expect to see significant improvements with increased model and data scale, more thoroughly cleaned data, as well as with alignment to human preference via instruction tuning or explicit optimization for preferences.

We expect that the rise of multimodal medical few-shot learners will lead to exciting opportunities with regard to model explainability (via rationale generation) as well as grounding the model in

verified sources (via multimodal retrieval to augment the few-shot prompt). Thereby, our work serves as a first step towards more generalist medical AI models (Moor et al., 2023a).

Limitations This work demonstrates a proof-of-concept. As such, Med-Flamingo is *not* intended nor safe for clinical use. In all VLMs we analyzed, hallucinations were observed. Furthermore, as Med-Flamingo is a pre-trained model without further instruction or preference tuning, it is possible that the model occasionally outputs low-quality generations.

Future work It will be an exciting route for future work to further train Med-Flamingo on clinical data, high-resolution medical image datasets as well as 3D volumes and medical videos. While current general-purpose medical VLMs are pre-trained on the broad medical literature (*i.e.,* they are only "book-smart"), also learning from diverse patient data directly will become crucial for down-stream applications.

7.7 Supplementary

7.7.1 Details for MTB dataset

Clustering the images In a post-hoc analysis, we clustered the image embeddings of the MTB dataset into a large number of clusters (100) and manually reviewed examples of each cluster to assign an annotation. We discard noisy or unclear clusters and display the remaining clusters and their frequency in Figure 7.6.

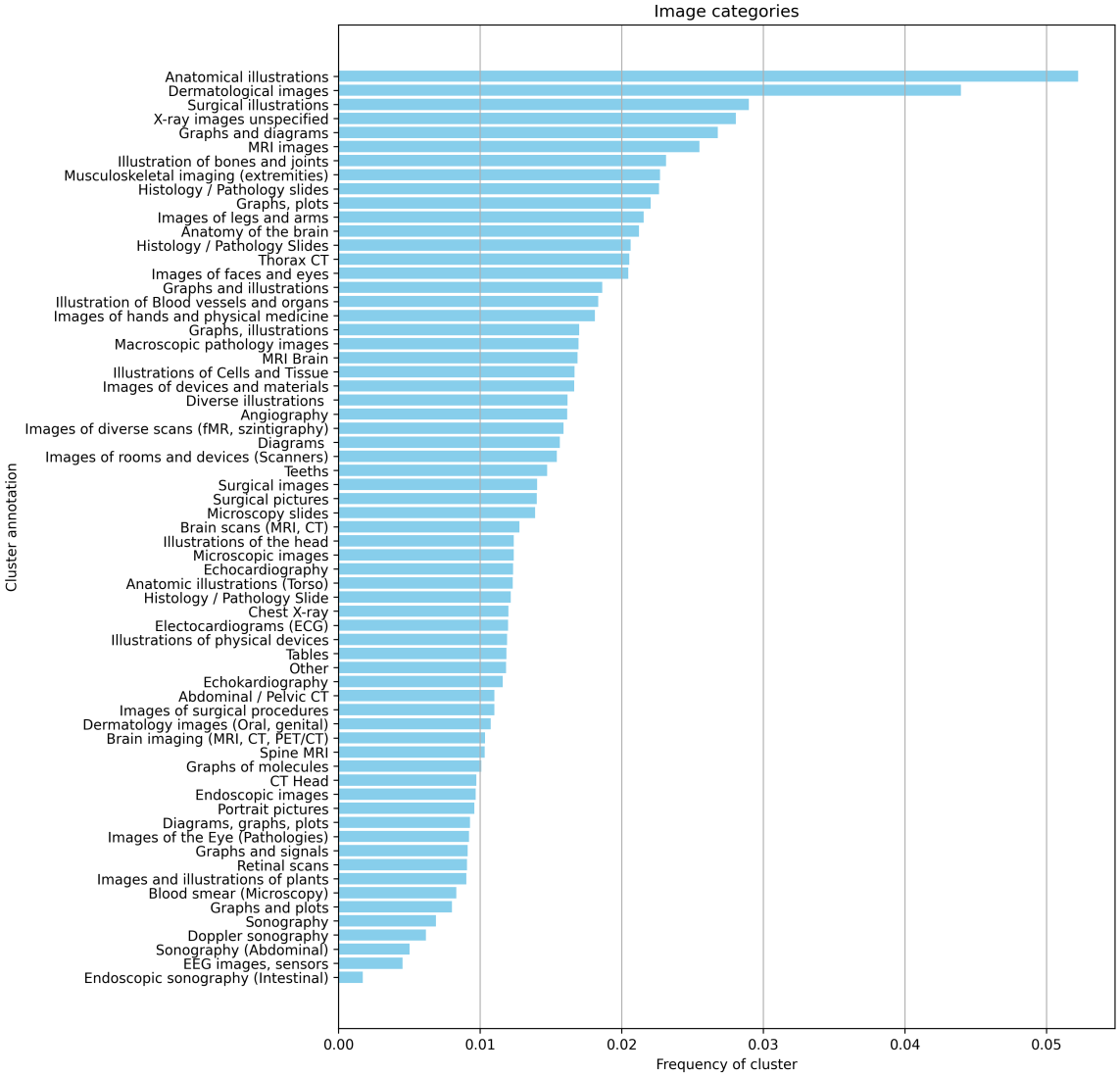


Figure 7.6: Distribution of manually annotated image clusters in the MTB dataset.

Classification of book titles Here, we provide further details about the creation of Figure 7.3. Table 7.2 lists the categories used to prompt the Claude-1 model to classify each book title. We initially prompted with 3 more very rare categories (Geriatrics, Occupational medicine, Space medicine), but merge them into the "Other" group for visualization purposes.

Neuroscience/Neurology	Obstetrics and Gynecology	Infectious Diseases
Radiology	Dermatology	Family medicine
Oncology	Immunology	Biomedical engineering
Surgery	Dentistry / Orthodontics	Anesthesiology
Cardiology	Ophthalmology	Physiology
Psychiatry	Pediatrics	Medical history
Pharmacology	Pathology	Nursing
Herbal medicine	Anatomy	Otolaryngology
Orthopedics	Gastroenterology	Hematology
Nutrition	Endocrinology	Urology
Internal Medicine	Genetics	Pulmonology
Sports Medicine	Medical Research and Statistics	Emergency Medicine
Cell Biology and Histology	Pain medicine	Public Health and Epidemiology
Forensics	Biochemistry	Nephrology
Critical care medicine	Medical Ethics	Veterinary medicine
Physical Medicine and Rehabilitation	Health informatics	Mindfulness
Other		

Table 7.2: List of 49 Categories (and "Other") used for visualing the MTB dataset in Figure 7.3

7.7.2 Details for Visual USMLE dataset

Figure 7.7 shows the distribution of specialty topics among the problems of the Visual USMLE dataset. Again, we used Claude-1 to classify each problem into categories provided in Table 7.2.

In Figure 7.8, we display an example problem of the Visual USMLE dataset.

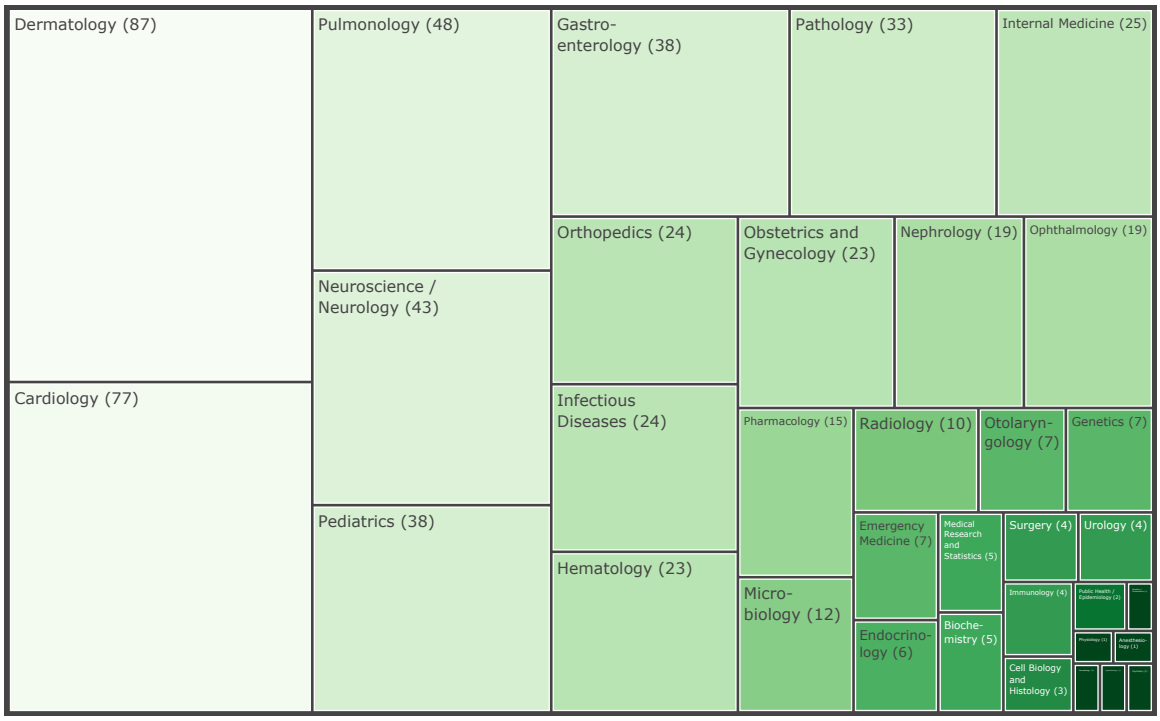


Figure 7.7: Distribution of specialty topics in the Visual USMLE dataset, as classified by Claude-1 using the categories provided in Table 7.2.

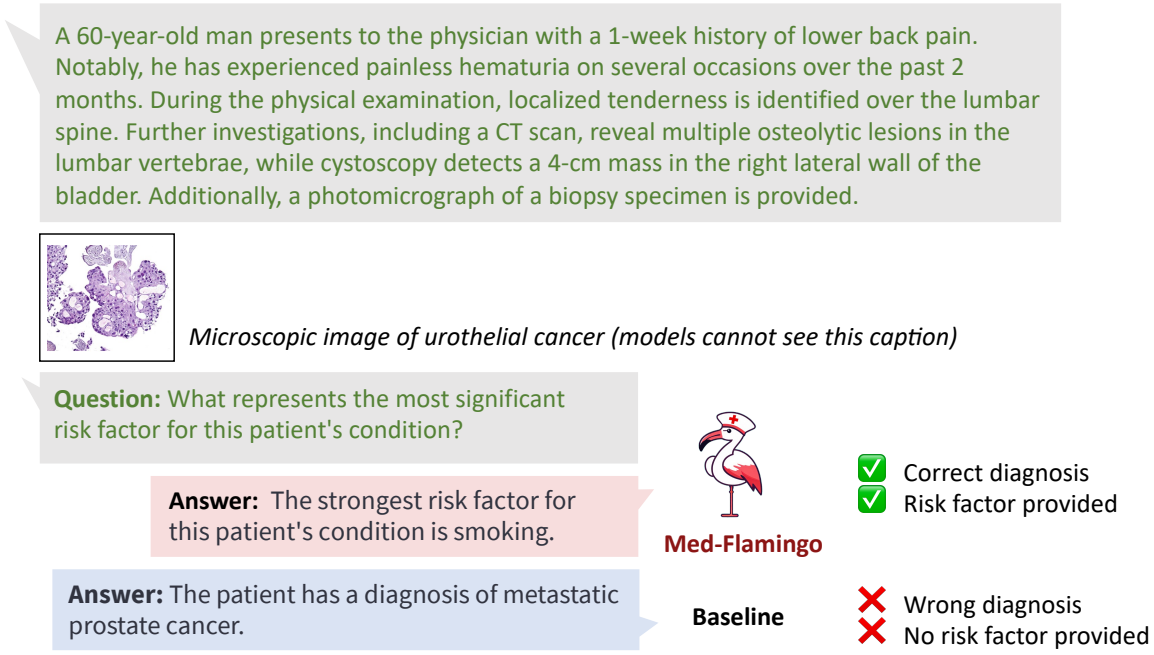


Figure 7.8: Example of a Visual USMLE problem. The displayed baseline answer is from the OpenFlamingo model.

Chapter 8

Conclusion

8.1 Contributions and impacts

Language models, such as GPT-4, have the capability to generate textual responses to user queries and find applications like question answering. However, to create truly versatile AI assistants, these models need to tackle more diverse and complex tasks involving domain or visual knowledge, such as answering medical questions and generating images. This necessitates accessing multimodal knowledge sources beyond text, such as knowledge bases and images.

In this thesis, we presented methods to build language models capable of using multimodal knowledge, including textual knowledge, structured knowledge, and visual knowledge, to perform diverse tasks for users (Part I). We then presented practical applications of these methods in medical scenarios, building clinical trial and question answering systems (Part II).

Specifically, in Chapter 3, we developed methods to **fuse textual knowledge** in models. Textual data provides broad and contextually-rich knowledge. Our developed method trains language models on a sequence of multiple related documents, facilitating the learning of broader and more complex knowledge from textual data. We show that this technique enhances long-context and multi-step reasoning performance of language models. Our trained language model excels particularly in tasks related to scientific text understanding and reasoning, which often require complex and extensive domain knowledge. Since its release in 2021, our model has consistently ranked as the top-performing system on the BLURB biomedical NLP leaderboard (Gu et al., 2021) up to the present day (2024). Moreover, our findings are inspiring subsequent research in the field, leading to the adoption of retrieval-augmented training and multi-document training in larger-scale language model training and their broader applications in biomedicine and healthcare (Shi et al., 2023a; Frisoni et al., 2022).

In Chapter 4, we developed methods to **fuse structured knowledge** in models. Structured knowledge graphs offer additional information and domain-specific knowledge to complement text. We introduced techniques that enable language models to leverage knowledge graph information.

Specifically, we devised a novel model architecture—a hybrid of language models and graph neural networks—to integrate knowledge graphs with textual data. Additionally, we proposed a training objective that facilitates joint reasoning across these two modalities. Our method surpasses existing language models in various natural language processing (NLP) tasks, particularly those involving domain-specific knowledge such as medical question answering. Notably, it achieved a new state-of-the-art accuracy in answering questions from the US medical licensing exam. In essence, our research has demonstrated that knowledge graphs can offer complementary information to textual data and can also serve as scaffolds for complex, multi-hop reasoning. This insight is inspiring subsequent works in the field, which leverage the strengths of both textual data and knowledge graphs across various applications (Sun et al., 2022; Wang et al., 2022a;b). In particular, as detailed in Chapter 6, we demonstrated that the fusion of textual knowledge and structured knowledge graphs enables the challenging medical application of clinical trial outcome prediction.

In Chapter 5, we developed methods to **fuse visual knowledge** in models. Concretely, we developed the first multimodal language model capable of retrieving and generating interleaved text and images. Our key technique is the unified model architecture designed to retrieve, fuse, and generate textual and visual elements using token representations. We demonstrate that this model not only enhances generation accuracy but also enables novel multimodal in-context learning and prompting capabilities. Our model is inspiring subsequent research in the field, such as extending and scaling the unified model architecture to include additional modalities like speech (Aghajanyan et al., 2023) and to perform a broader range of downstream tasks (Yu et al., 2023).

In Chapter 6, we presented the **application** of our knowledge fusion techniques to **clinical trial outcome prediction**. Motivated by the current inefficiencies and high costs associated with clinical trials, we developed a model to forecast the safety and efficacy of a given clinical trial in advance, leveraging trial documents and clinical knowledge graphs. We show that the model, trained on historical clinical trial data, can accurately predict outcomes for new, previously unseen trials. This suggests the potential to reduce costs and enhance safety and efficacy in future clinical trials. In Chapter 7, we presented the **application to multimodal medical question answering**. The increased demand for healthcare motivates the development of fast and accurate interfaces for healthcare providers and patients, such as medical question answering systems. Given the diverse nature of medical data, encompassing text and images such as X-rays, we apply the technique of textual and visual knowledge fusion to build a medical QA system that can handle multimodal content. Our system demonstrates a significant (20%) improvement in clinical usefulness compared to prior medical QA systems, as evaluated by clinical experts. These two aforementioned applications demonstrate the practical and wide-ranging impact of our knowledge fusion approaches in domains such as medicine and healthcare.

8.2 Future directions for multimodal models

8.2.1 Modeling

In this thesis, we have developed a model capable of integrating text, knowledge graphs, and images. The future goal is to build a truly unified and versatile model that encompasses all modalities in both input and output, accommodating any common form of data such as text, speech/audio, images, videos, graphs, tables, databases, gene expressions, and more. Furthermore, the input and output of such a model should ideally accommodate any *mixed* sequence of modalities, for instance, a prompt consisting of text, images, and videos. This model will enable a broader spectrum of practical applications, including video editing based on textual and image prompts (e.g., "please edit this video [video] by adding this scenery [image of the scenery] in the background") or generating radiology reports using X-ray images and medical databases.

To expand this thesis and build truly unified multimodal models, the following research questions and challenges need to be addressed as the next steps:

- Efficient representation of different modalities: Tokenizing every modality—text, audio, image, video, graphs, databases, etc.—is a plausible approach (Esser et al., 2021; Arnab et al., 2021; Hsu et al., 2021; Kim et al., 2022). However, a significant challenge lies in the tokenizers' consumption of tokens for non-textual modalities. For instance, recent image tokenizers like VQGAN (Esser et al., 2021) require numerous tokens for each image (e.g., 1000 tokens), and a similar issue arises for videos (Arnab et al., 2021), audio/speech (Hsu et al., 2021), and graph (Kim et al., 2022) modalities. Developing improved tokenizers capable of compressing each modality into fewer tokens while enhancing reconstruction quality is crucial. Additionally, determining the appropriate token consumption for non-textual modalities, considering their information density relative to text, is another relevant research question.

- Scaling with increased modalities: As more modalities are incorporated, the sequence length (i.e., number of tokens) will naturally increase. Consequently, there is a need to devise efficient models capable of processing long sequences of information computationally efficiently while maintaining strong task performance.

8.2.2 Alignment

In this thesis, our focus has been on either the pre-training of models or their application to specific tasks, such as clinical trials. The alignment process is an important next step for making multimodal models more widely useful for human users. For instance, envision a conversational system akin to ChatGPT capable of performing various tasks but with the ability to use more modalities than just text.

The alignment process is crucial for transforming language models into practically useful systems like ChatGPT due to the disparity between the raw pre-trained model behavior and user intent and context. Techniques such as instruction tuning (Wei et al., 2021) and learning from human feedback (Ouyang et al., 2022; Rafailov et al., 2023) are important for fine-tuning models for various tasks and domains, ensuring their outputs align with user instructions, expectations, and preferences, including practical utility like multi-turn conversation.

For multimodal models, such as vision-language models, this alignment process becomes even more critical and challenging. With modalities beyond text, including image, video, and audio, the scope of potential applications expands significantly (e.g., image generation/editing, video generation/editing), necessitating greater care to ensure model outputs align with human preferences, societal norms, and to mitigate harmful biases or inappropriate content. To achieve this goal, we think that curating high-quality data for instruction tuning and human preference in multimodal models is a key next step. Efforts are currently being made towards it (Yu et al., 2023), such as collecting various vision-language tasks like visual question answering, object detection, image editing, and image style transfer.

8.2.3 Evaluation

Benchmarks and evaluations drive advancements in AI by guiding the community toward areas for improvement (Deng et al., 2009; Rajpurkar et al., 2016; Ethayarajh and Jurafsky, 2020; Raji et al., 2021). In this thesis, our main focus lied in evaluating models based on the accuracy of their outputs. However, a crucial next step for advancing multimodal models is to establish a more rigorous and holistic evaluation framework and benchmarks.

For instance, in the NLP and language model research, the adoption of meta-benchmarks (Wang et al., 2018; Koh et al., 2021; Srivastava et al., 2022; Qin et al., 2023) and holistic evaluations (Liang et al., 2022) across various scenarios (such as question answering and creative writing) and aspects (such as accuracy and bias) has enabled comprehensive assessments of model capabilities and limitations in both technical and societal aspects. This approach has accelerated the development of models that perform well across diverse aspects and cater to a broader range of users.

Holistic and rigorous evaluations are particularly crucial for multimodal models. Firstly, as modalities extend from text to visuals such as images and videos, there is an increased need to evaluate a wider range of application tasks and aspects, necessitating a more comprehensive evaluation approach. For example, beyond accuracy, aspects such as aesthetics (e.g., important for applications in art and design), originality (e.g., concerning copyright issues), safety (e.g., identifying illegal or inappropriate content), and bias (e.g., ensuring demographic representation) become vital considerations in visual generation tasks.

Secondly, certain aspects, like aesthetics and originality in visual generation, can be challenging to evaluate rigorously as they are subjective and difficult to define precisely. Therefore, automated

evaluation metrics alone may not suffice, necessitating human evaluation.

In our ongoing work (Lee et al., 2023), we are developing holistic benchmarks and evaluation frameworks for multimodal models, starting with text-to-image generation models (e.g., DALL-E (Ramesh et al., 2022), Stable Diffusion (Rombach et al., 2022)). Specifically, we consider 12 diverse aspects crucial in image generation: text-image alignment, image quality (realism), aesthetics, originality, reasoning, knowledge, bias, toxicity, fairness, robustness, multilinguality, and efficiency. We evaluate the latest text-to-image models across these 12 aspects using both automated metrics and human-rated metrics to uncover the current capabilities and limitations of models and guide future model development.

Future directions include evaluating more general vision-language models (e.g., GPT-4 vision (OpenAI, 2023), Gemini (Team et al., 2023), and LLaVA (Liu et al., 2023)), and subsequently to multimodal models incorporating additional modalities such as audio and video.

Bibliography

Sylvia Adams, Megan Othus, Sandip Pravin Patel, Kathy D Miller, Rashmi Chugh, Scott M Schuetze, Mary D Chamberlin, Barbara J Haley, Anna Maria V Storniolo, Mridula P Reddy, et al. A multicenter phase II trial of ipilimumab and nivolumab in unresectable or metastatic metaplastic breast cancer: Cohort 36 of dual anti–CTLA-4 and anti–PD-1 blockade in rare tumors (dart, swog s1609) ipilimumab and nivolumab in rare tumors s1609: Metaplastic. *Clinical Cancer Research*, 28(2):271–278, 2022.

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. Htlm: Hyper-text pre-training and prompting of language models. *arXiv preprint arXiv:2107.06955*, 2021.

Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. CM3: A causal masked multimodal model of the internet. *arXiv preprint arXiv:2201.07520*, 2022.

Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. Scaling laws for generative mixed-modal language models. *arXiv preprint arXiv:2301.03728*, 2023.

Adel Ahmed, Naser Al-Masri, Yousef S Abu Sultan, Alaa N Akkila, Abdelbaset Almasri, Ahmed Y Mahmoud, Ihab S Zaqout, and Samy S Abu-Naser. Knowledge-based systems survey, 2019.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual

language model for few-shot learning. In *Advances in Neural Information Processing Systems*, 2022a. URL https://openreview.net/forum?id=EbMuimAbPbs.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022b.

Zarqa Ali, John Robert Zibert, and Simon Francis Thomsen. Virtual clinical trials: Perspectives in dermatology. *Dermatology*, 236(4):375–382, 2020.

David Ansari, Daniel Ansari, Roland Andersson, and Åke Andrén-Sandberg. Pancreatic cancer and thromboembolic disease, 150 years after trousseau. *Hepatobiliary surgery and nutrition*, 4(5):325, 2015.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations (ICLR)*, 2020.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

Euan A Ashley. Towards precision medicine. *Nature Reviews Genetics*, 17(9):507–522, 2016.

Oron Ashual, Shelly Sheynin, Adam Polyak, Uriel Singer, Oran Gafni, Eliya Nachmani, and Yaniv Taigman. Knn-diffusion: Image generation via large-scale retrieval. *arXiv preprint arXiv:2204.02849*, 2022.

Nir Atias and Roded Sharan. An algorithmic framework for predicting side effects of drugs. *Journal of Computational Biology*, 18(3), 2011.

Anas Awadalla, Irena Gao, Joshua Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo, March 2023. URL https://doi.org/10.5281/zenodo.7733589.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Simon Baker, Ilona Silins, Yufan Guo, Imran Ali, Johan Högberg, Ulla Stenius, and Anna Korhonen. Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics*, 2016.

Andrew L Beam, Benjamin Kompa, Allen Schmaltz, Inbar Fried, Griffin Weber, Nathan Palmer, Xu Shi, Tianxi Cai, and Isaac S Kohane. Clinical concept embeddings learned from massive sources of multimodal medical data. In *Pacific Symposium on Biocomputing 2020*, pages 295–306. World Scientific, 2019.

Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: Pretrained language model for scientific text. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

Adam Berger, Stephen A Della Pietra, and Vincent J Della Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996.

Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. Content-based citation recommendation. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. Abductive commonsense reasoning. In *International Conference on Learning Representations (ICLR)*, 2020.

Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. Multimodal datasets: misogyny, pornography, and malignant stereotypes. *arXiv preprint arXiv:2110.01963*, 2021a.

Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. Ethical considerations of generative AI. *AI for Content Creation Workshop, CVPR*, 2021b.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*, 2020.

Andreas Blattmann, Robin Rombach, Kaan Oktay, and Björn Ommer. Retrieval-augmented diffusion models. *arXiv preprint arXiv:2204.11824*, 2022.

Olivier Bodenreider. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic acids research*, 2004.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, 2008.

Elliot Bolton, David Hall, Michihiro Yasunaga, Tony Lee, Chris Manning, and Percy Liang. BioMedLM: a domain-specific large language model for biomedical text, 2022. URL `https://crfm.stanford.edu/2022/12/15/biomedlm.html`.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NeurIPS)*, 2013.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning (ICML)*, 2022.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. In *Association for Computational Linguistics (ACL)*, 2019.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

Àlex Bravo, Janet Piñero, Núria Queralt-Rosinach, Michael Rautschka, and Laura I Furlong. Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research. *BMC bioinformatics*, 2015.

Maria Brbic, Michihiro Yasunaga, Prabhat Agarwal, and Jure Leskovec. Predicting drug outcome of population via clinical knowledge graph, 2024.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Elliot G Brown, Louise Wood, and Sue Wood. The medical dictionary for regulatory activities (meddra). *Drug safety*, 20(2):109–117, 1999.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Lynell Burmark. *Visual Literacy: Learn To See, See To Learn.* ERIC, 2002.

Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew E Peters, Arie Cattan, and Ido Dagan. Cross-document language modeling. In *Findings of EMNLP*, 2021.

Iacer Calixto, Alessandro Raganato, and Tommaso Pasini. Wikipedia entities as rendezvous across languages: Grounding multilingual language models by predicting wikipedia hyperlinks. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. In *International Workshop on Semantic Evaluation (SemEval)*, 2017.

M Chalabi, A Cardona, DR Nagarkar, A Dhawahir Scala, DR Gandara, A Rittmeyer, ML Albert, T Powles, M Kok, FG Herrera, et al. Efficacy of chemotherapy and atezolizumab in patients with non-small-cell lung cancer receiving antibiotics and proton pump inhibitors: pooled post hoc analyses of the oak and poplar trials. *Annals of Oncology*, 31(4):525–531, 2020.

Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. Pre-training tasks for embedding-based large-scale retrieval. In *International Conference on Learning Representations (ICLR)*, 2020.

Hyasmine Charles, Chester B Good, Barbara H Hanusa, Chung-Chou H Chang, and Jeff Whittle. Racial differences in adherence to cardiac medications. *Journal of the National Medical Association*, 95(1):17, 2003.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. In *Association for Computational Linguistics (ACL)*, 2017.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*, 2016.

Rui Chen, George I Mias, Jennifer Li-Pook-Than, Lihua Jiang, Hugo YK Lam, Rong Chen, Elana Miriami, Konrad J Karczewski, Manoj Hariharan, Frederick E Dewey, et al. Personal omics profiling reveals dynamic molecular and medical phenotypes. *Cell*, 148(6):1293–1307, 2012.

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W. Cohen. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2022a.

Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. Re-imagen: Retrieval-augmented text-to-image generator. *arXiv preprint arXiv:2209.14491*, 2022b.

Feixiong Cheng, Rishi J Desai, Diane E Handy, Ruisheng Wang, Sebastian Schneeweiss, Albert-Laaszlo Barabasi, and Joseph Loscalzo. Network-based approach to prediction and population-based validation of in silico drug repurposing. *Nature Communications*, 9(1):1–12, 2018.

Feixiong Cheng, István A Kovács, and Albert-László Barabási. Network-based prediction of drug combinations. *Nature Communications*, 10(1):1–11, 2019.

Jaemin Cho, Jiasen Lu, Dustin Schwenk, Hannaneh Hajishirzi, and Aniruddha Kembhavi. X-lxmert: Paint, caption and answer questions with multi-modal transformers. *arXiv preprint arXiv:2009.11278*, 2020.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Dawn R Cochrane, Sebastián Bernales, Britta M Jacobsen, Diana M Cittelly, Erin N Howe, Nicholas C D'Amato, Nicole S Spoelstra, Susan M Edgerton, Annie Jean, Javier Guerrero, et al. Role of the androgen receptor in breast cancer and preclinical analysis of enzalutamide. *Breast Cancer Research*, 16:1–19, 2014.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. Specter: Document-level representation learning using citation-informed transformers. In *Association for Computational Linguistics (ACL)*, 2020.

Michael Collins. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637, 2003.

Gene Ontology Consortium. The Gene Ontology resource: enriching a GOld mine. *Nucleic Acids Research*, 49(D1):D325–D334, 2021a.

UniProt Consortium. UniProt: The universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 2021b.

Paolo Curatolo, David N Franz, John A Lawson, Zuhal Yapici, Hiroko Ikeda, Tilman Polster, Rima Nabbout, Petrus J de Vries, Dennis J Dlugos, Jenna Fan, et al. Adjunctive everolimus for children and adolescents with treatment-refractory seizures associated with tuberous sclerosis complex: post-hoc analysis of the phase 3 exist-3 trial. *The Lancet Child & Adolescent Health*, 2(7):495–504, 2018.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, 2005.

Thomas Davenport and Ravi Kalakota. The potential for artificial intelligence in healthcare. *Future healthcare journal*, 6(2):94, 2019.

Allan Peter Davis, Thomas C Wiegers, Jolene Wiegers, Robin J Johnson, Daniela Sciaky, Cynthia J Grondin, and Carolyn J Mattingly. Chemical-induced phenotypes at CTD help inform the predisease state and construct adverse outcome pathways. *Toxicological Sciences*, 165(1):145–156, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021.

Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *arXiv preprint arXiv:2204.14217*, 2022.

Luc Y Dirix, Istvan Takacs, Guy Jerusalem, Petros Nikolinakos, Hendrik-Tobias Arkenau, Andres Forero-Torres, Ralph Boccia, Marc E Lippman, Robert Somer, Martin Smakal, et al. Avelumab, an anti-PD-L1 antibody, in patients with locally advanced or metastatic breast cancer: a phase 1b JAVELIN solid tumor study. *Breast Cancer Research and Treatment*, 167:671–686, 2018.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 2014.

William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*, 2017.

William B Ershler. Capecitabine monotherapy: safe and effective treatment for metastatic breast cancer. *The Oncologist*, 11(4):325–335, 2006.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Kawin Ethayarajh and Dan Jurafsky. Utility is in the eye of the user: A critique of nlp leaderboards. *arXiv preprint arXiv:2009.13888*, 2020.

Antonio Fabregat, Steven Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, et al. The reactome pathway knowledgebase. *Nucleic acids research*, 46(D1):D649–D655, 2018.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Yannick Djoumbou Feunang, Roman Eisner, Craig Knox, Leonid Chepelev, Janna Hastings, Gareth Owen, Eoin Fahy, Christoph Steinbeck, Shankar Subramanian, Evan Bolton, et al. ClassyFire: automated chemical classification with a comprehensive, computable taxonomy. *Journal of Cheminformatics*, 8(1):1–20, 2016.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. In *Workshop on Machine Reading for Question Answering*, 2019.

AI Forever. rudall-e. `https://github.com/ai-forever/ru-dalle`, 2021.

Flavia Franconi, Sandra Brunelleschi, Luca Steardo, and Vincenzo Cuomo. Gender differences in drug responses. *Pharmacological Research*, 55(2):81–95, 2007.

Daniel Fried, Armen Aghajanyan, Jessy Lin, Sida Wang, Eric Wallace, Freda Shi, Ruiqi Zhong, Wen-tau Yih, Luke Zettlemoyer, and Mike Lewis. Incoder: A generative model for code infilling and synthesis. *arXiv preprint arXiv:2204.05999*, 2022.

Andrea Friesenhengst, Tamara Pribitzer-Winner, Heidi Miedl, Katharina Pröstling, and Martin Schreiber. Elevated aromatase (CYP19A1) expression is associated with a poor survival of patients with estrogen receptor positive breast cancer. *Hormones and Cancer*, 9(2):128–138, 2018.

Giacomo Frisoni, Miki Mizutani, Gianluca Moro, and Lorenzo Valgimigli. Bioreader: a retrieval-enhanced text-to-text transformer for biomedical literature. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 5770–5793, 2022.

Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. *arXiv preprint arXiv:2203.13131*, 2022.

Diego Galeano, Shantao Li, Mark Gerstein, and Alberto Paccanaro. Predicting the frequencies of drug side effects. *Nature Communications*, 11(1):1–14, 2020.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Association for Computational Linguistics (ACL)*, 2021.

Rodolfo Garza-Morales, Roxana Gonzalez-Ramos, Akiko Chiba, Roberto Montes de Oca-Luna, Lacey R McNally, Kelly M McMasters, and Jorge G Gomez-Gutierrez. Temozolomide enhances triple-negative breast cancer virotherapy in vitro. *Cancers*, 10(5):144, 2018.

Paul Geels, Elizabeth Eisenhauer, Andrea Bezjak, Benny Zee, and Andrew Day. Palliative effect of chemotherapy: objective tumor response is associated with symptom improvement in patients with metastatic breast cancer. *Journal of Clinical Oncology*, 18(12):2395–2405, 2000.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings of EMNLP*, 2020.

Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 1983.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, 2007.

Carlotta Giani, Laura Valerio, Alberto Bongiovanni, Cosimo Durante, Giorgio Grani, Toni Ibrahim, Stefano Mariotti, Michela Massa, Fabiana Pani, Gabriella Pellegriti, et al. Safety and quality-of-life data from an Italian expanded access program of lenvatinib for treatment of thyroid cancer. *Thyroid*, 31(2):224–232, 2021.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1): 1–23, 2021.

Shir Gur, Natalia Neverova, Chris Stauffer, Ser-Nam Lim, Douwe Kiela, and Austin Reiter. Cross-modal retrieval augmentation for multi-modal classification. *arXiv preprint arXiv:2104.08108*, 2021.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. In *International Conference on Machine Learning (ICML)*, 2020.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

Omid Hamid, Robert Ilaria Jr, Claus Garbe, Pascal Wolter, Michele Maio, Thomas E Hutson, Ana Arance, Paul Lorigan, Jeeyun Lee, Axel Hauschild, et al. A randomized, open-label clinical trial of tasisulam sodium versus paclitaxel as second-line treatment in patients with metastatic melanoma. *Cancer*, 120(13):2016–2024, 2014.

Will Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. Embedding logical queries on knowledge graphs. *Advances in neural information processing systems*, 31, 2018.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

HS Han, Véronique Diéras, M Robson, M Palácová, PK Marcom, Agnes Jager, I Bondarenko, D Citrin, Mario Campone, ML Telli, et al. Veliparib with temozolomide or carboplatin/paclitaxel versus placebo with carboplatin/paclitaxel in patients with BRCA1/2 locally recurrent/metastatic breast cancer: randomized phase II study. *Annals of Oncology*, 29(1):154–161, 2018.

Yaru Hao, Haoyu Song, Li Dong, Shaohan Huang, Zewen Chi, Wenhui Wang, Shuming Ma, and Furu Wei. Language models are general-purpose interfaces. *arXiv preprint arXiv:2206.06336*, 2022.

Daniel M Hartung, Deborah A Zarin, Jeanne-Marie Guise, Marian McDonagh, Robin Paynter, and Mark Helfand. Reporting discrepancies between the clinicaltrials.gov results database and peer-reviewed publications. *Annals of Internal Medicine*, 160(7):477–483, 2014.

Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. A retrieve-and-edit framework for predicting structured outputs. In *Neural Information Processing Systems (NeurIPS)*, 2018.

Frederick Hayes-Roth, Donald A Waterman, and Douglas B Lenat. *Building expert systems*. Addison-Wesley Longman Publishing Co., Inc., 1983.

Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. Integrating graph contextualized knowledge into pre-trained language models. In *Findings of EMNLP*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

Gary G Hendrix, Earl D Sacerdoti, Daniel Sagalowicz, and Jonathan Slocum. Developing a natural language interface to complex data. *ACM Transactions on Database Systems (TODS)*, 3(2): 105–147, 1978.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

Sam Henry, Kevin Buchan, Michele Filannino, Amber Stubbs, and Ozlem Uzuner. 2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records. *Journal of the American Medical Informatics Association*, 27(1):3–12, 2020.

Monika R Henzinger, Allan Heydon, Michael Mitzenmacher, and Marc Najork. On near-uniform url sampling. *Computer Networks*, 2000.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of biomedical informatics*, 2013.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Neural Information Processing Systems (NeurIPS)*, 2017.

Micheal Hewett, Diane E Oliver, Daniel L Rubin, Katrina L Easton, Joshua M Stuart, Russ B Altman, and Teri E Klein. Pharmgkb: the pharmacogenetics knowledge base. *Nucleic acids research*, 30(1):163–165, 2002.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2020.

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

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019b.

Rachael P Huntley, Tony Sawford, Prudence Mutowo-Meullenet, Aleksandra Shypitsyna, Carlos Bonilla, Maria J Martin, and Claire O'Donovan. The GOA database: Gene Ontology Annotation updates for 2015. *Nucleic Acids Research*, 43(D1):D1057–D1063, 2015.

Cheng Jin, Heng Yu, Jia Ke, Peirong Ding, Yongju Yi, Xiaofeng Jiang, Xin Duan, Jinghua Tang, Daniel T Chang, Xiaojian Wu, et al. Predicting treatment response from longitudinal images using multi-task deep learning. *Nature Communications*, 12(1):1851, 2021a.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 2021b.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 2019.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Association for Computational Linguistics (ACL)*, 2017.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 2020.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. Jointgt: Graph-text joint representation learning for text generation from knowledge graphs. In *Findings of ACL*, 2021.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

Anita Khadka, Ivan Cantador, and Miriam Fernandez. Exploiting citation knowledge in personalised recommendation of recent scientific publications. In *Language Resources and Evaluation Conference (LREC)*, 2020.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations (ICLR)*, 2019.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. Unifiedqa: Crossing format boundaries with a single qa system. In *Findings of EMNLP*, 2020.

Bosung Kim, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. Multi-task learning for knowledge graph completion with pre-trained language models. In *International Conference on Computational Linguistics (COLING)*, 2020.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, 2004.

Jinwoo Kim, Tien Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, and Seunghoon Hong. Pure transformers are powerful graph learners. *arXiv*, abs/2207.02505, 2022. URL https://arxiv.org/abs/2207.02505.

Saehoon Kim, Sanghun Cho, Chiheon Kim, Doyup Lee, and Woonhyuk Baek. mindall-e on conceptual captions. https://github.com/kakaobrain/minDALL-E, 2021a.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Research*, 49(D1):D1388–D1395, 2021b.

Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Darrell G Kirch and Kate Petelle. Addressing the physician shortage: the peril of ignoring demography. *Jama*, 317(19):1947–1948, 2017.

Dani Kiyasseh, Runzhuo Ma, Taseen F Haque, Brian J Miles, Christian Wagner, Daniel A Donoho, Animashree Anandkumar, and Andrew J Hung. A vision transformer for decoding surgeon activity from surgical videos. *Nature Biomedical Engineering*, pages 1–17, 2023.

Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting of the association for computational linguistics*, pages 423–430, 2003.

Todd C Knepper and Howard L McLeod. When will clinical trials finally reflect diversity? *Nature*, 2018.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

Jun Kong, Lee AD Cooper, Fusheng Wang, David A Gutman, Jingjing Gao, Candace Chisolm, Ashish Sharma, Tony Pan, Erwin G Van Meir, Tahsin M Kurc, et al. Integrative, multimodal analysis of glioblastoma using tcga molecular data, pathology images, and clinical outcomes. *IEEE Transactions on Biomedical Engineering*, 58(12):3469–3474, 2011.

Martin Krallinger, Obdulia Rabal, Saber A Akhondi, Martın Pérez Pérez, Jesús Santamaría, Gael Pérez Rodríguez, Georgios Tsatsaronis, and Ander Intxaurrondo. Overview of the biocreative vi chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics (TACL)*, 2019.

Adam Lavertu and Russ B Altman. Redmed: Extending drug lexicons for social media applications. *Journal of biomedical informatics*, 99:103307, 2019.

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*, 2020.

Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Benita Teufel, Marco Bellagente, et al. Holistic evaluation of text-to-image models. *arXiv preprint arXiv:2311.04287*, 2023.

Douglas B Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

John P Leonard, Marek Trneny, Koji Izutsu, Nathan H Fowler, Xiaonan Hong, Jun Zhu, Huilai Zhang, Fritz Offner, Adriana Scheliga, Grzegorz S Nowakowski, et al. Augment: a phase iii study of lenalidomide plus rituximab versus placebo plus rituximab in relapsed or refractory indolent lymphoma. *Journal of Clinical Oncology*, 37(14):1188, 2019.

Hector J Levesque. Knowledge representation and reasoning. *Annual review of computer science*, 1 (1):255–287, 1986.

Yoav Levine, Noam Wies, Daniel Jannai, Dan Navon, Yedid Hoshen, and Amnon Shashua. The inductive bias of in-context learning: Rethinking pretraining example design. *arXiv preprint arXiv:2110.04541*, 2021.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. Pre-training via paraphrasing. In *Neural Information Processing Systems (NeurIPS)*, 2020a.

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*, 2020b.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020c.

Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. *Neural Information Processing Systems (NeurIPS)*, 2019a.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *arXiv preprint arXiv:2306.00890*, 2023.

Da Li, Sen Yang, Kele Xu, Ming Yi, Yukai He, and Huaimin Wang. Multi-task pre-training language model for semantic network completion. *arXiv preprint arXiv:2201.04843*, 2022a.

Irene Li, Michihiro Yasunaga, Muhammed Yavuz Nuzumlalı, Cesar Caraballo, Shiwani Mahajan, Harlan Krumholz, and Dragomir Radev. A neural topic-attention model for medical term abbreviation disambiguation. *Machine Learning for Health (ML4H)*, 2019b.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Yehao Li, Yingwei Pan, Ting Yao, and Tao Mei. Comprehending and ordering semantics for image captioning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022b.

Percy Liang, Michael I Jordan, and Dan Klein. Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446, 2013.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. In *Findings of ACL*, 2021.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, 2014.

Weixiong Lin, Ziheng Zhao, Xiaoman Zhang, Chaoyi Wu, Ya Zhang, Yanfeng Wang, and Weidi Xie. Pmc-clip: Contrastive language-image pre-training using biomedical documents. *arXiv preprint arXiv:2303.07240*, 2023.

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

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. Self-alignment pretraining for biomedical entity representations. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

Katherine A Liu and Natalie A Dipietro Mager. Women's involvement in clinical trials: historical perspective and future implications. *Pharmacy Practice (Granada)*, 14(1):0–0, 2016.

Mei Liu, Yonghui Wu, Yukun Chen, Jingchun Sun, Zhongming Zhao, Xue-wen Chen, Michael Edwin Matheny, and Hua Xu. Large-scale prediction of adverse drug reactions using chemical, biological, and phenotypic properties of drugs. *Journal of the American Medical Informatics Association*, 19 (e1):e28–e35, 2012.

Ruishan Liu, Shemra Rizzo, Samuel Whipple, Navdeep Pal, Arturo Lopez Pineda, Michael Lu, Brandon Arnieri, Ying Lu, William Capra, Ryan Copping, et al. Evaluating eligibility criteria of oncology trials using real-world data and AI. *Nature*, 592(7855):629–633, 2021b.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and P. Wang. K-bert: Enabling language representation with knowledge graph. In *AAAI Conference on Artificial Intelligence*, 2020.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

GV Long, KT Flaherty, D Stroyakovskiy, H Gogas, E Levchenko, F De Braud, J Larkin, C Garbe, T Jouary, A Hauschild, et al. Dabrafenib plus trametinib versus dabrafenib monotherapy in patients with metastatic BRAF V600E/K-mutant melanoma: long-term survival and safety analysis of a phase 3 study. *Annals of Oncology*, 28(7):1631–1639, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6):bbac409, 2022.

Yunan Luo, Xinbin Zhao, Jingtian Zhou, Jinglin Yang, Yanqing Zhang, Wenhua Kuang, Jian Peng, Ligong Chen, and Jianyang Zeng. A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nature Communications*, 8(1):1–13, 2017.

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Zhengyi Ma, Zhicheng Dou, Wei Xu, Xinyu Zhang, Hao Jiang, Zhao Cao, and Ji-Rong Wen. Pretraining for ad-hoc retrieval: Hyperlink is also you need. In *Conference on Information and Knowledge Management (CIKM)*, 2021.

Eric Margolis, Stephen Laurence, et al. *Concepts: core readings*. Mit Press, 1999.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

John McCoy, Andy Goren, Flávio Adsuara Cadegiani, Sergio Vaño-Galván, Maja Kovacevic, Mirna Situm, Jerry Shapiro, Rodney Sinclair, Antonella Tosti, Andrija Stanimirovic, et al. Proxalutamide reduces the rate of hospitalization for COVID-19 male outpatients: A randomized double-blinded placebo-controlled trial. *Frontiers in Medicine*, page 1043, 2021.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *Journal of Open Source Software*, 3(29):861, 2018.

Ninareh Mehrabi, Pei Zhou, Fred Morstatter, Jay Pujara, Xiang Ren, and A. G. Galstyan. Lawyers are dishonest? quantifying representational harms in commonsense knowledge resources. *ArXiv*, abs/2103.11320, 2021.

Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. Rethinking search: making domain experts out of dilettantes. In *ACM SIGIR Forum*, 2021.

Todor Mihaylov and Anette Frank. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Association for Computational Linguistics (ACL)*, 2018.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Scott Miller, David Stallard, Robert Bobrow, and Richard Schwartz. A fully statistical approach to natural language interfaces. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 55–61, 1996.

Pooya Mobadersany, Safoora Yousefi, Mohamed Amgad, David A Gutman, Jill S Barnholtz-Sloan, José E Velázquez Vega, Daniel J Brat, and Lee AD Cooper. Predicting cancer outcomes from histology and genomics using convolutional networks. *Proceedings of the National Academy of Sciences*, 115(13):E2970–E2979, 2018.

Michael Moor, Oishi Banerjee, Zahra Shakeri Hossein Abad, Harlan M Krumholz, Jure Leskovec, Eric J Topol, and Pranav Rajpurkar. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259–265, 2023a.

Michael Moor, Qian Huang, Shirley Wu, Michihiro Yasunaga, Yash Dalmia, Jure Leskovec, Cyril Zakka, Eduardo Pontes Reis, and Pranav Rajpurkar. Med-flamingo: a multimodal medical few-shot learner. In *Machine Learning for Health (ML4H)*, pages 353–367. PMLR, 2023b.

Stuart J Nelson, Kelly Zeng, John Kilbourne, Tammy Powell, and Robin Moore. Normalized names for clinical drugs: RxNorm at 6 years. *Journal of the American Medical Informatics Association*, 18(4):441–448, 2011.

Anastasios Nentidis, Konstantinos Bougiatiotis, Anastasia Krithara, and Georgios Paliouras. Results of the seventh edition of the bioasq challenge. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2019.

Allen Newell and Herbert A Simon. Computer science as empirical inquiry: Symbols and search. In *ACM Turing award lectures*, page 1975, 2007.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain J Marshall, Ani Nenkova, and Byron C Wallace. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, 2018.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.

National Library of Medicine. PubMed. https://pubmed.ncbi.nlm.nih.gov/, 1996.

Dan Ofer and Michal Linial. ProFET: Feature engineering captures high-level protein functions. *Bioinformatics*, 31(21):3429–3436, 2015.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. Unified open-domain question answering with structured and unstructured knowledge. *arXiv preprint arXiv:2012.14610*, 2020.

OpenAI. Gpt-4 technical report, 2023.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. Finding deceptive opinion spam by any stretch of the imagination. *arXiv preprint arXiv:1107.4557*, 2011.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Rushad Patell, Thomas Bogue, Anita Koshy, Poorva Bindal, Mwanasha Merrill, William C Aird, Kenneth A Bauer, and Jeffrey I Zwicker. Postdischarge thrombosis and hemorrhage in patients with COVID-19. *Blood*, 136(11):1342–1346, 2020.

Azabelle Peters and Prasanna Tadi. Aromatase inhibitors. *StatPearls [Internet]*, 2021.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Matthew E. Peters, Mark Neumann, IV RobertLLogan, Roy Schwartz, V. Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Roberto Pieraccini, Esther Levin, and Chin-Hui Lee. Stochastic representation of conceptual structure in the atis task. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*, 1991.

Janet Piñero, Àlex Bravo, Núria Queralt-Rosinach, Alba Gutiérrez-Sacristán, Jordi Deu-Pons, Emilio Centeno, Javier García-García, Ferran Sanz, and Laura I Furlong. Disgenet: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Research*, page gkw943, 2016.

Janet Piñero, Juan Manuel Ramírez-Anguita, Josep Saüch-Pitarch, Francesco Ronzano, Emilio Centeno, Ferran Sanz, and Laura I Furlong. The DisGeNET knowledge platform for disease genomics: 2019 update. *Nucleic Acids Research*, 2019.

Franco Piovella, Luciano Crippa, Marisa Barone, S Vigano D'Angelo, Silvia Serafini, Laura Galli, Chiara Beltrametti, and Armando D'Angelo. Normalization rates of compression ultrasonography in patients with a first episode of deep vein thrombosis of the lower limbs: association with recurrence and new thrombosis. *Haematologica*, 87(5):515–522, 2002.

Andrew P Prayle, Matthew N Hurley, and Alan R Smyth. Compliance with mandatory reporting of clinical trial results on clinicaltrials. gov: cross sectional study. *BMJ*, 344, 2012.

Sudeep Pushpakom, Francesco Iorio, Patrick A Eyers, K Jane Escott, Shirley Hopper, Andrew Wells, Andrew Doig, Tim Guilliams, Joanna Latimer, Christine McNamee, et al. Drug repurposing: progress, challenges and recommendations. *Nature Reviews Drug Discovery*, 18(1):41–58, 2019.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. TimeML: Robust specification of event and temporal expressions in text. *New Directions in Question Answering*, 3:28–34, 2003.

Vahed Qazvinian and Dragomir R Radev. Scientific paper summarization using citation summary networks. In *International Conference on Computational Linguistics (COLING)*, 2008.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*, 2023.

Dragomir R Radev, Hong Qi, Harris Wu, and Weiguo Fan. Evaluating web-based question answering systems. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2002.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763, 2021a. URL https://proceedings.mlr.press/v139/radford21a.html. ISSN: 2640-3498.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021b.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

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 (JMLR)*, 2020.

Inioluwa Deborah Raji, Emily M Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna. Ai and the everything in the whole wide world benchmark. *arXiv preprint arXiv:2111.15366*, 2021.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

A Ramamoorthy, MA Pacanowski, J Bull, and L Zhang. Racial/ethnic differences in drug disposition and response: review of recently approved drugs. *Clinical Pharmacology & Therapeutics*, 97(3): 263–273, 2015.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning (ICML)*, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Rita Ramos, Bruno Martins, Desmond Elliott, and Yova Kementchedjhieva. Smallcap: Lightweight image captioning prompted with retrieval augmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Christian Raymond and Giuseppe Riccardi. Generative and discriminative algorithms for spoken language understanding. In *Interspeech 2007-8th Annual Conference of the International Speech Communication Association*, 2007.

Hongyu Ren and Jure Leskovec. Beta embeddings for multi-hop logical reasoning in knowledge graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Hongyu Ren, Weihua Hu, and Jure Leskovec. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In *International Conference on Learning Representations (ICLR)*, 2020.

Hongyu Ren, Hanjun Dai, Bo Dai, Xinyun Chen, Michihiro Yasunaga, Haitian Sun, Dale Schuurmans, Jure Leskovec, and Denny Zhou. Lego: Latent execution-guided reasoning for multi-hop question answering on knowledge graphs. In *International Conference on Machine Learning (ICML)*, 2021.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation extraction with matrix factorization and universal schemas. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2013.

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Mark Robson, Seock-Ah Im, Elżbieta Senkus, Binghe Xu, Susan M Domchek, Norikazu Masuda, Suzette Delaloge, Wei Li, Nadine Tung, Anne Armstrong, et al. Olaparib for metastatic breast cancer in patients with a germline BRCA mutation. *New England Journal of Medicine*, 377(6): 523–533, 2017.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Joseph S Ross, Gregory K Mulvey, Elizabeth M Hines, Steven E Nissen, and Harlan M Krumholz. Trial publication after registration in clinicaltrials. gov: a cross-sectional analysis. *PLoS Medicine*, 6(9):e1000144, 2009.

Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. Knowledge-aware language model pretraining. *arXiv preprint arXiv:2007.00655*, 2020.

Hope S Rugo, Olufunmilayo I Olopade, Angela DeMichele, Christina Yau, Laura J van't Veer, Meredith B Buxton, Michael Hogarth, Nola M Hylton, Melissa Paoloni, Jane Perlmutter, et al. Adaptive randomization of veliparib–carboplatin treatment in breast cancer. *New England Journal of Medicine*, 375(1):23–34, 2016.

Camilo Ruiz, Marinka Zitnik, and Jure Leskovec. Identification of disease treatment mechanisms through the multiscale interactome. *Nature communications*, 12(1):1–15, 2021.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Yoshinobu Saito, Mikie Nagayama, Yukiko Miura, Satoko Ogushi, Yasutomo Suzuki, Rintaro Noro, Yuji Minegishi, Go Kimura, Yukihiro Kondo, and Akihiko Gemma. A case of pneumocystis pneumonia associated with everolimus therapy for renal cell carcinoma. *Japanese Journal of Clinical Oncology*, 43(5):559–562, 2013.

Claude Sammut and Geoffrey I. Webb, editors. *TF–IDF*, pages 986–987. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8.

Alberto Santos, Ana R Colaço, Annelaura B Nielsen, Lili Niu, Maximilian Strauss, Philipp E Geyer, Fabian Coscia, Nicolai J Wewer Albrechtsen, Filip Mundt, Lars Juhl Jensen, et al. A knowledge graph to interpret clinical proteomics data. *Nature Biotechnology*, 40(5):692–702, 2022.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Sara Sarto, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Retrieval-augmented transformer for image captioning. In *Proceedings of the 19th International Conference on Content-based Multimedia Indexing*, 2022.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

Nicholas J Schork. Personalized medicine: time for one-person trials. *Nature*, 520(7549):609–611, 2015.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

Yeon Seonwoo, Sang-Woo Lee, Ji-Hoon Kim, Jung-Woo Ha, and Alice Oh. Weakly supervised pre-training for multi-hop retriever. In *Findings of ACL*, 2021.

Tao Shen, Yi Mao, Pengcheng He, Guodong Long, Adam Trischler, and Weizhu Chen. Exploiting structured knowledge in text via graph-guided representation learning. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. Towards controllable biases in language generation. In *the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)-Findings, long*, 2020.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Victoria Lin, Noah A Smith, Luke Zettlemoyer, Scott Yih, and Mike Lewis. In-context pretraining: Language modeling beyond document boundaries. *arXiv preprint arXiv:2310.10638*, 2023a.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023b.

Karus Siegel, D Karus, and EW Schrimshaw. Racial differences in attitudes toward protease inhibitors among older HIV-infected men. *AIDS care*, 12(4):423–434, 2000.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, Philip Mansfield, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. Large language models encode clinical knowledge, 2022. URL http://arxiv.org/abs/2212.13138.

Larry Smith, Lorraine K Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, et al. Overview of biocreative ii gene mention recognition. *Genome biology*, 2008.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

Gizem Soğancıoğlu, Hakime Öztürk, and Arzucan Özgür. Biosses: a semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 2017.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958, 2014.

Syracuse Post Standard. Speakers give sound advice. *Syracuse Post Standard*, 28(18):1, March 1911.

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.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

Carole H Sudre, Benjamin Murray, Thomas Varsavsky, Mark S Graham, Rose S Penfold, Ruth C Bowyer, Joan Capdevila Pujol, Kerstin Klaser, Michela Antonelli, Liane S Canas, et al. Attributes and predictors of long COVID. *Nature medicine*, 27(4):626–631, 2021.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W Cohen. Open domain question answering using early fusion of knowledge bases and text. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Haitian Sun, Tania Bedrax-Weiss, and William W Cohen. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019a.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuan-Jing Huang, and Zheng Zhang. Colake: Contextualized language and knowledge embedding. In *International Conference on Computational Linguistics (COLING)*, 2020.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*, 2021.

Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2022.

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 (ICLR)*, 2019b.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

Amol Takalkar, Bhavna Paryani, Scott Adams, and Vivek Subbiah. Radium-223 dichloride therapy in breast cancer with osseous metastases. *Case Reports*, 2015:bcr2015211152, 2015.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Eve Tang, Philippe Ravaud, Carolina Riveros, Elodie Perrodeau, and Agnes Dechartres. Comparison of serious adverse events posted at clinicaltrials.gov and published in corresponding journal articles. *BMC Medicine*, 13(1):1–8, 2015.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja,

Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Martin Chadwick, Gaurav Singh Tomar, Xavier Garcia, Evan Senter, Emanuel Taropa, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Yujing Zhang, Ravi Addanki, Antoine Miech, Annie Louis, Laurent El Shafey, Denis Teplyashin, Geoff Brown, Elliot Catt, Nithya Attaluri, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaly Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, Hanzhao Lin, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yong Cheng, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia,

James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, YaGuang Li, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Gamaleldin Elsayed, Ed Chi, Mahdis Mahdieh, Ian Tenney, Nan Hua, Ivan Petrychenko, Patrick Kane, Dylan Scandinaro, Rishub Jain, Jonathan Uesato, Romina Datta, Adam Sadovsky, Oskar Bunyan, Dominik Rabiej, Shimu Wu, John Zhang, Gautam Vasudevan, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Betty Chan, Pam G Rabinovitch, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Sahitya Potluri, Jane Park, Elnaz Davoodi, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Chris Gorgolewski, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Paul Suganthan, Evan Palmer, Geoffrey Irving, Edward Loper, Manaal Faruqui, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Michael Fink, Alfonso

Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marin Georgiev, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Minnie Lui, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Lam Nguyen Thiet, Daniel Andor, Pedro Valenzuela, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Sarmishta Velury, Sebastian Krause, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Tejasi Latkar, Mingyang Zhang, Quoc Le, Elena Allica Abellan, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Sid Lall, Ken Franko, Egor Filonov, Anna Bulanova, Rémi Leblond, Vikas Yadav, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Hao Zhou, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Jeremiah Liu, Mark Omernick, Colton Bishop, Chintu Kumar, Rachel Sterneck, Ryan Foley, Rohan Jain, Swaroop Mishra, Jiawei Xia, Taylor Bos, Geoffrey Cideron, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Petru Gurita, Hila Noga, Premal Shah, Daniel J. Mankowitz, Alex Polozov, Nate Kushman, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Anhad Mohananey, Matthieu Geist, Sidharth Mudgal, Sertan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Quan Yuan, Sumit Bagri, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Aliaksei Severyn, Jonathan Lai, Kathy Wu, Heng-Tze Cheng, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Mark Geller, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Andrei Sozanschi, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Abhimanyu Goyal, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Sabaer Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Tao Zhu, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Dustin Tran, Yeqing Li, Nir Levine, Ariel Stolovich, Norbert Kalb, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Balaji Lakshminarayanan, Charlie Deck, Shyam Upadhyay, Hyo Lee, Mike Dusenberry, Zonglin Li, Xuezhi Wang, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Summer Yue, Sho Arora, Eric Malmi,

Daniil Mirylenka, Qijun Tan, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Steven Zheng, Francesco Pongetti, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Ragha Kotikalapudi, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Chenkai Kuang, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Pei Sun, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Ishita Dasgupta, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Yuan Liu, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Ivo Penchev, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Adam Kurzrok, Lynette Webb, Sahil Dua, Dong Li, Preethi Lahoti, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Taylan Bilal, Evgenii Eltyshev, Daniel Balle, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Adams Yu, Christof Angermueller, Xiaowei Li, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Kevin Brooks, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Komal Jalan, Dinghua Li, Ginger Perng, Blake Hechtman, Parker Schuh, Milad Nasr, Mia Chen, Kieran Milan, Vladimir Mikulik, Trevor Strohman, Juliana Franco, Tim Green, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2023.

Ekin Tiu, Ellie Talius, Pujan Patel, Curtis P Langlotz, Andrew Y Ng, and Pranav Rajpurkar. Expert-level detection of pathologies from unannotated chest x-ray images via self-supervised learning. *Nature Biomedical Engineering*, 6(12):1399–1406, 2022.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 human language*

technology conference of the north american chapter of the association for computational linguistics, pages 252–259, 2003.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Tiffany A Traina, Kathy Miller, Denise A Yardley, Janice Eakle, Lee S Schwartzberg, Joyce O'Shaughnessy, William Gradishar, Peter Schmid, Eric Winer, Catherine Kelly, et al. Enzalutamide for the treatment of androgen receptor–expressing triple-negative breast cancer. *Journal of clinical oncology*, 36(9):884, 2018.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. In *Workshop on Representation Learning for NLP*, 2017.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning (ICML)*, 2016.

ME Trudeau, M Crump, D Charpentier, L Yelle, L Bordeleau, S Matthews, and E Eisenhauer. Temozolomide in metastatic breast cancer (MBC): a phase II trial of the National Cancer Institute of Canada–Clinical Trials Group (NCIC-CTG). *Annals of Oncology*, 17(6):952–956, 2006.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*, 2019.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems (NeurIPS)*, 2017.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 2014.

Adrienne G Waks and Eric P Winer. Breast cancer treatment: a review. *JAMA*, 321(3):288–300, 2019.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018.

Hongwei Wang, Hongyu Ren, and Jure Leskovec. Relational message passing for knowledge graph completion. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021a.

Kuan Wang, Yuyu Zhang, Diyi Yang, Le Song, and Tao Qin. Gnn is a counter? revisiting gnn for question answering. In *International Conference on Learning Representations (ICLR)*, 2022a.

Phil Wang. Dall-e in pytorch. `https://github.com/lucidrains/DALLE-pytorch`, 2021.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics (TACL)*, 2021b.

Yanan Wang, Michihiro Yasunaga, Hongyu Ren, Shinya Wada, and Jure Leskovec. Vqa-gnn: Reasoning with multimodal semantic graph for visual question answering. *arXiv preprint arXiv:2205.11501*, 2022b.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations (ICLR)*, 2022c.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 2019.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39:165–210, 2005.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2017.

William E Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage., 1990.

Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.

David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 2018.

William Woods. The lunar sciences natural language information system. *BBN report*, 1972.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *International Conference on Learning Representations (ICLR)*, 2020.

Hua Xu, Shane P Stenner, Son Doan, Kevin B Johnson, Lemuel R Waitman, and Joshua C Denny. MedEx: a medication information extraction system for clinical narratives. *Journal of the American Medical Informatics Association*, 17(1):19–24, 2010.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? In *International Conference on Learning Representations (ICLR)*, 2020.

Yichong Xu, Chenguang Zhu, Shuohang Wang, Siqi Sun, Hao Cheng, Xiaodong Liu, Jianfeng Gao, Pengcheng He, Michael Zeng, and Xuedong Huang. Human parity on commonsenseqa: Augmenting self-attention with external attention. In *Association for Computational Linguistics (ACL)*, 2022.

Yiwen Xu, Ahmed Hosny, Roman Zeleznik, Chintan Parmar, Thibaud Coroller, Idalid Franco, Raymond H Mak, and Hugo JWL Aerts. Deep learning predicts lung cancer treatment response from serial medical imaginglongitudinal deep learning to track treatment response. *Clinical Cancer Research*, 25(11):3266–3275, 2019.

Jun Yan, Mrigank Raman, Aaron Chan, Tianyu Zhang, Ryan Rossi, Handong Zhao, Sungchul Kim, Nedim Lipka, and Xiang Ren. Learning contextualized knowledge structures for commonsense reasoning. In *Findings of ACL*, 2021.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Association for Computational Linguistics (ACL)*, 2019.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations (ICLR)*, 2015.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. Graph-based neural multi-document summarization. In *Conference on Computational Natural Language Learning (CoNLL)*, 2017.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D. Manning, Percy Liang, and Jure Leskovec. Deep bidirectional language-knowledge graph pretraining. In *Neural Information Processing Systems (NeurIPS)*, 2022a.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. LinkBERT: Pretraining language models with document links. In *Association for Computational Linguistics (ACL)*, 2022b.

Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Rich James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Retrieval-augmented multimodal language modeling. In *International Conference on Machine Learning (ICML)*, 2023.

Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating explanations for graph neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. Jaket: Joint pre-training of knowledge graph and language understanding. In *AAAI Conference on Artificial Intelligence*, 2022a.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022b.

Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2023.

Chi Yuan, Patrick B Ryan, Casey Ta, Yixuan Guo, Ziran Li, Jill Hardin, Rupa Makadia, Peng Jin, Ning Shang, Tian Kang, et al. Criteria2query: a natural language interface to clinical databases for cohort definition. *Journal of the American Medical Informatics Association*, 26(4):294–305, 2019.

John M Zelle and Raymond J Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055, 1996.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Association for Computational Linguistics (ACL)*, 2019.

Luke S Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *arXiv preprint arXiv:1207.1420*, 2012.

Sheng Zhang, Yanbo Xu, Naoto Usuyama, Jaspreet Bagga, Robert Tinn, Sam Preston, Rajesh Rao, Mu Wei, Naveen Valluri, Cliff Wong, et al. Large-scale domain-specific pretraining for biomedical vision-language processing. *arXiv preprint arXiv:2303.00915*, 2023a.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022a.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with bert. In *International Conference on Learning Representations (ICLR)*, 2020.

Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. Pmc-vqa: Visual instruction tuning for medical visual question answering. *arXiv preprint arXiv:2305.10415*, 2023b.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. GreaseLM: Graph reasoning enhanced language models for question answering. In *International Conference on Learning Representations (ICLR)*, 2022b.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. In *Association for Computational Linguistics (ACL)*, 2019.

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 *International Conference on Computer Vision (ICCV)*, 2015.

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