Factual Knowledge-enhanced Question Answering in Dynamic Environments

by

Sagi Shaier

B.S., Kennesaw State University, 2018

M.S., University of Colorado Boulder, 2023

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2025

Committee Members:

Katharina von der Wense, Chair

Lawrence E. Hunter

Matt Jones

James H. Martin

Maria L. Pacheco

Shaier, Sagi (Ph.D., Computer Science)

Factual Knowledge-enhanced Question Answering in Dynamic Environments

Thesis directed by Prof. Katharina von der Wense

This thesis explores the realm of question answering (QA) systems, with a focus on those that leverage external knowledge to enhance their capabilities. While standard QA systems have demonstrated success, their integration with external knowledge unlocks new potentials, enabling them to reason over novel or conflicting information beyond their parametric knowledge. However, existing systems still face significant challenges, such as inaccuracies due to a lack of factual grounding, limited adaptability to dynamic environments. This research addresses these shortcomings and others by proposing innovative approaches that elevate the efficacy and reliability of QA systems.

Central to this work is the development of QA systems capable of handling a broad range of topics, modalities, and knowledge representations, with an emphasis on factual accuracy and contextual reasoning. Such systems hold immense potential to assist large populations, particularly in high-stakes domains like biomedicine, where individuals often face fear, confusion, and barriers to accessing affordable, quality healthcare. By providing accurate and actionable responses, these systems can empower users to make informed decisions in urgent situations.

The research progresses through three core areas: knowledge probing, knowledge usage, and model improvement. In Knowledge Probing, novel methods are developed to assess what models have learned internally, particularly in text-based and KG-grounded systems, given the significance of parametric knowledge in shaping model outputs. In Knowledge Usage, research focuses on characterizing how models leverage internal and external knowledge when answering questions, shedding light on their decision-making processes. Finally, in Model Improvement, methodologies are designed to address issues identified in prior research, enhancing QA systems' performance, reliability, and factuality based on established desiderata.

Acknowledgements

First, I would like to thank Meghan Burke, who introduced me to research. Meghan, without your kindness, I would never have found my way to Tsetse flies, pigs, and epidemiology, nor would I have ended up in Australia, hugging kangaroos in the wild—or become a researcher. You also introduced me to sweet potato casserole, which is arguably just as life-changing. So, in many ways, both my academic journey and my expanded palate are entirely your fault.

Secondly, I would like to thank Glen Meades, who instilled in me the importance of being detail oriented. Meades, you taught me to focus on the tiniest details—because, as you often reminded me through many memorable exams, a single word in a sentence can completely change its meaning. You have also taught me how to think creatively about random mathematical problems, as well as many useful English words, many of which I still can't pronounce.

Last but foremost, I would like to thank my advisors Katharina Von Der Wense and Lawrence Hunter, who welcomed me into their labs and continuously mentored me throughout my PhD.

Katharina, your incredible attention to detail has taught me how to read and write papers more effectively, better focus my research, and think critically about problems. You've also introduced me to fun German slang, snorkeling, and scuba diving, for which I'm extremely grateful.

Larry, you've supported me throughout most of my degree, both with funding and as an incredible mentor in research and life. Thank you for being a steady source of support, helping me stay calm through many failures, and offering invaluable guidance along the way.

I've learned so much from both of you, and I truly believe it has made me a better researcher and person. You have become not just mentors, but also close friends.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

Humans have always been driven by curiosity, constantly seeking answers to the myriad questions that arise in our minds. This innate thirst for knowledge has fueled centuries of exploration, discovery, and innovation. Yet, in the vast expanse of human inquiry, the sheer volume and complexity of information can often overwhelm even the most diligent seekers.

In today's digital age, where information is readily accessible through the internet, books, and other sources, one might assume that finding answers to questions would be easier than ever before. However, the reality is far more nuanced. While the wealth of information available is unparalleled, navigating this sea of data to uncover precise, relevant answers remains a formidable challenge.

Many questions require not only a deep understanding of the subject matter but also the ability to synthesize information from diverse sources, ranging from textual documents to graphical representations. Moreover, the complexity of these questions varies widely, making it difficult for individuals across different domains to comprehend and respond effectively. For instance, deciphering complex medical or physics texts often proves daunting for those not well-versed in these specialized fields.

In light of these challenges, there arises a pressing need for intelligent systems capable of efficiently answering questions. These systems, known as question answering (QA) systems, offer a promising avenue for streamlining the information retrieval process and providing users with accurate, comprehensive responses.

The focus of this thesis is to explore and enhance such QA systems, with a specific emphasis

on leveraging external knowledge sources. By integrating these rich repositories of information into QA systems, I aim to enhance their ability to comprehend and respond to a wide range of queries, spanning various domains and levels of complexity.

Through rigorous analysis, experimentation, and innovation, this research endeavors to contribute to the advancement of QA systems, ultimately empowering users with more efficient and reliable means of accessing the wealth of human knowledge available at their fingertips.

In the chapters that follow, we will delve into the theoretical foundations of QA systems, examine existing approaches and methodologies, propose novel techniques for incorporating external knowledge sources, and evaluate the performance of these enhanced systems through empirical studies and real-world applications. By engaging in this exploration, I strive to pave the way for a future where accessing and understanding information is not only effortless, but truly transformative.

Chapter 2

Background

2.1 Language Models and Their Role in Question Answering

2.1.1 Definition of Language Models

Language models (LMs) play a crucial role in natural language processing (NLP) by estimating the probabilities of different linguistic units within a specific context [119]. These units can range from individual symbols to entire token sequences, encompassing words, phrases, and more. More concretely, LMs estimate the probability of a symbol S, such as a part of a word (i.e., a token), given the context of the previous $n-1$ symbols. This estimation is encapsulated by the expression $P(S \mid context)$, where *context* refers to the sequence of the preceding $n-1$ symbols.

Essentially, LMs provide computational frameworks for analyzing and predicting the structure and probability distribution of language, enabling machines to understand, generate, and interact with human language in a manner that closely resembles natural communication. Through extensive training on vast text data, LMs acquire the ability to comprehend and produce coherent and contextually appropriate language, making them essential tools in various applications such as machine translation, sentiment analysis, and question answering.

2.1.2 Evolution of Language Models

Traditional LMs, such as n-gram models [119], rely on fixed-window contexts and suffered from sparsity issues. These models assume that the probability of a word occurring could be

approximated by the frequencies of n-grams observed in a training corpus. While effective for short-range dependencies, this approach failed to account for the complexities of long-range dependencies, leading to poor generalization in larger corpora. The advent of neural networks represented a significant leap forward, enabling more powerful architectures that could learn richer representations of language. Notable early neural architectures included recurrent neural networks (RNNs) [70], which introduced a method of maintaining hidden states across time steps, and long short-term memory networks (LSTMs) [97], which alleviated the vanishing gradient problem associated with RNNs, allowing for better learning of long-term dependencies. However, these architectures still faced limitations in terms of computational efficiency and their ability to capture very long-range dependencies due to their inherently sequential processing nature.

The introduction of the Transformer model [296] marked a paradigm shift, moving away from recurrent connections to attention-based mechanisms. This new architecture allowed for efficient parallelized processing of sequences, enabling the capture of long-range dependencies without the sequential bottleneck inherent in RNNs and LSTMs. By leveraging self-attention, the Transformer could consider all tokens in a sequence simultaneously and learn context-dependent relationships more effectively. This was a significant improvement over previous architectures that struggled with tasks requiring long-term contextual understanding.

Modern LMs are predominantly based on the Transformer architecture, which employs self-attention mechanisms and feed-forward layers to process entire sequences in parallel. Key components of the Transformer include:

Self-Attention: Self-attention allows the model to assign different importance weights to tokens within a sequence, enabling it to capture long-range dependencies and contextual relationships that RNNs and LSTMs often struggled with due to their sequential processing constraints. In particular, self-attention operates on the entire sequence at once, making it possible to learn interactions between all tokens regardless of their positions within the sequence.

Positional Encoding: Since the attention mechanism does not inherently account for the sequential order of tokens, positional encoding is introduced to inject position-specific information

into the model. This allows the model to differentiate token positions and retain the notion of sequence order, an essential feature for understanding sentence structure and syntactic relationships.

Multi-Head Attention: Multi-head attention allows the model to simultaneously attend to different parts of the sequence with multiple attention heads, each learning a different aspect of the input sequence. This enhances the model's ability to capture diverse types of relationships and improve the quality of the learned representations.

The Transformer differs from traditional sequential models, such as RNNs, in several important ways:

Parallelization: Unlike RNNs, which process sequences one token at a time in a sequential manner, the Transformer model handles all sequence elements simultaneously. This parallelization enables much faster training and inference, as the model can process large batches of data at once without being constrained by the need to process tokens in order.

Long-Range Dependencies: RNNs and LSTMs struggle to capture long-range dependencies due to the vanishing gradient problem, which can prevent them from learning relationships between tokens that are far apart in a sequence. In contrast, Transformers excel at capturing these long-range dependencies by using self-attention, which allows them to directly model the relationships between distant tokens.

Scalability: Transformers can accommodate variable-length input sequences without requiring padding, enhancing flexibility and scalability compared to fixed-length architectures like simple multi-layer perceptrons (MLPs). Additionally, the architecture's ability to scale efficiently with increasing data and model size has been a driving force behind its widespread adoption in modern NLP.

Today, there are many Transformer variants. However, several models have gained widespread recognition and influence in the field:

- **BERT (Bidirectional Encoder Representations from Transformers)** [56]: Developed by Google AI in 2018, BERT is a pre-trained Transformer model that has achieved

state-of-the-art results across a wide range of NLP tasks, such as question answering, sentiment analysis, and named entity recognition. BERT is trained using a masked language modeling (MLM) task, where a percentage of input tokens are randomly masked, and the model is tasked with predicting these masked tokens based on their context. This bidirectional pre-training enables BERT to capture both left and right context simultaneously, giving it a deeper understanding of language.

- **GPT (Generative Pre-trained Transformer) Series** [28]: Developed by OpenAI, the GPT series includes models like GPT, GPT-2, and GPT-3. These are autoregressive models that generate text by predicting the next token in a sequence given the previous context. Unlike BERT, which uses a MLM objective, GPT models are trained to predict the next token in a sentence, making them particularly well-suited for text generation and other sequence-based tasks.

- **T5 (Text-to-Text Transfer Transformer)** [229]: Introduced by Google AI, T5 is a versatile Transformer model that treats every NLP task as a text-to-text problem. By framing tasks like translation, summarization, and question answering as text generation problems, T5 simplifies the model architecture and training process. This unified approach allows T5 to be applied to a wide range of tasks with a single model, making it more flexible and easier to deploy across different NLP applications.

2.1.3 Prominent Architectures

Broadly, modern LMs can be categorized into three architectual types: encoder-only, decoder-only, and encoder-decoder models. Each has its distinct structure and use cases in NLP.

2.1.3.1 Encoder-Only Models

Encoder-only models primarily focus on processing and encoding input sequences. These models are designed to capture the rich contextual representations of input data, which can then be

used for downstream tasks such as classification, token-level predictions, or extracting information.

A prominent example of encoder-only models is BERT, which uses a stack of Transformer encoder layers to process input sequences bidirectionally. The key strength of encoder-only models is their ability to efficiently capture the relationships between input tokens, making them highly effective for tasks where understanding the full context of the input sequence is crucial. However, they do not directly generate sequences, which limits their use for text generation tasks.

2.1.3.2 Decoder-Only Models

Decoder-only models are typically used in tasks that require sequence generation, such as QA, text completion, translation, and summarization. In these models, the decoder processes previously generated tokens to predict the next token in the sequence, making them autoregressive in nature.

A widely recognized example of decoder-only models is the GPT series. These models rely on a stack of Transformer decoder layers, where each token in the sequence is generated one after another. The advantage of decoder-only models is their ability to generate fluent and diverse text, as they are trained to predict the next token in an autoregressive manner. However, they typically lack bidirectional context understanding, which can affect their performance on tasks requiring deep comprehension of the entire input sequence.

2.1.3.3 Encoder-Decoder Models

Encoder-decoder models combine the benefits of both the encoder and decoder architectures, making them suitable for tasks that require both understanding and generation. These models use the encoder to process and represent the input sequence and the decoder to generate the output sequence based on the encoder's representation.

A famous example of encoder-decoder models is T5, which frames all NLP tasks as a unified text-to-text problem. In T5, the encoder processes input text, while the decoder generates output text. This approach simplifies the training and deployment process, as the model can be used for a wide range of tasks, including translation, summarization, and question answering, all within the

same framework.

Encoder-decoder models are highly flexible and powerful, as they can handle both complex understanding (through the encoder) and generation (through the decoder). They are particularly effective in tasks like machine translation, where the model needs to generate a sequence of tokens (the translated text) based on an input sequence (the source text).

The main advantage of encoder-decoder models is their versatility, enabling them to handle a wide range of tasks involving both input understanding and output generation. However, their complexity requires more computational resources and training time compared to simpler encoder-only or decoder-only models.

2.1.4 Training Paradigms

Contemporary LMs typically follow a two-stage training process: pretraining and fine-tuning [315]. These stages enable models to learn generalized language patterns and then adapt them to specific tasks or domains. The first stage, pretraining, involves training the model on large corpora of unlabeled text, often using self-supervised objectives. In the second stage, fine-tuning, the pretrained model is further adapted using labeled data for a specific task, allowing it to perform optimally on that task.

2.1.4.1 Pretraining

Pretraining serves as the foundation for contemporary models by enabling them to learn general language patterns and features from vast amounts of text. This stage typically involves unsupervised or self-supervised learning, where models are exposed to large, diverse datasets without the need for explicit task labels, which are expensive to create. The most commonly used pretraining objectives are *MLM* (used by BERT) and *autoregressive prediction* (used by GPT).

Masked Language Modeling (MLM): MLM, popularized by BERT [56], is a type of self-supervised learning where some percentage of input tokens are randomly masked, and the model is trained to predict the masked tokens based on their surrounding context. This objective enables

the model to learn bidirectional representations of the text, capturing context from both the left and right of each token. One of the advantages of MLM is its ability to model complex syntactic and semantic relationships by learning to understand the full context of a sentence or passage, not just the local dependencies.

For instance, in the sentence "The cat sat on the _," BERT would predict the missing word (e.g., "mat") by understanding both the surrounding words ("The," "cat," "sat," and "on") and the word itself in the context of the sentence. This bidirectional training approach allows the model to gain a more holistic understanding of language compared to traditional unidirectional models.

Autoregressive Language Modeling: In contrast to MLM, autoregressive language models, such as GPT [28], are trained to predict the next token in a sequence given the preceding tokens. The model is trained to maximize the likelihood of the next token in a sequence, conditioning on the previously generated tokens. This approach enables the model to generate coherent sequences of text and is particularly effective for tasks involving text generation, such as writing, dialogue systems, or code generation.

Autoregressive models like GPT are trained on large corpora of text, using a left-to-right processing order. This means the model only has access to past tokens when predicting the next one.

Transfer Learning in Pretraining: Pretraining also serves as a form of transfer learning. The pretrained model's weights encode knowledge from diverse and massive datasets, enabling it to transfer that knowledge to a wide range of downstream tasks. By leveraging large-scale, unsupervised data, pretrained models are able to learn language representations that are general enough to be useful across multiple applications, from machine translation to text summarization.

The size of the pretraining dataset plays a critical role in the performance of the model. More diverse datasets allow the model to learn richer, more robust representations of language, which can improve its performance on a variety of tasks. As a result, large language models (LLMs) are often trained on corpora containing billions of words, sourced from diverse domains such as books, websites, and academic papers.

2.1.4.2 Fine-tuning

Fine-tuning adapts a pretrained model to a specific task by training it on a labeled dataset that is task-specific. The process involves adjusting the parameters of the pretrained model to optimize performance on a specific objective, such as classification, question answering, or sentiment analysis. Fine-tuning allows models to leverage the general knowledge acquired during pretraining and apply it to specialized tasks, achieving high performance with relatively little labeled data.

Task-Specific Adaptation: During fine-tuning, a pretrained model is exposed to a dataset with input-output pairs relevant to the specific task. For example, in a question answering task, the model might be trained on a dataset where the input consists of a question and a passage of text, and the output is the correct answer span from the passage. By adjusting the model's weights during fine-tuning, it can specialize its language representations to solve the particular task effectively.

Fine-tuning typically uses supervised learning, where the model is provided with labeled examples. In contrast to pretraining, which uses unsupervised or self-supervised methods, fine-tuning ensures the model learns the specific patterns or features needed to solve the task at hand. The learning rate during fine-tuning is often smaller than during pretraining to avoid catastrophic forgetting, ensuring that the model retains the knowledge gained during pretraining while adapting to the new task.

Transfer of Knowledge: The advantage of fine-tuning is that it allows the model to transfer general knowledge learned during pretraining to the target task. For example, a model pretrained on vast amounts of text can quickly adapt to a specific domain, such as medical question answering, by fine-tuning on a smaller labeled dataset in the medical domain. This process greatly reduces the amount of task-specific data needed to achieve strong performance, making it especially useful for domains where labeled data is scarce.

Transfer Learning Techniques: Fine-tuning can be done in a variety of ways depending on the task and the size of the labeled dataset. Common strategies include:

- **Full Fine-Tuning:** All of the model's parameters are updated during the fine-tuning

process [166]. This approach is typically used when a large labeled dataset is available, as it allows the model to fully adapt to the task.

- **Layer-wise Fine-Tuning:** Only certain layers of the model are fine-tuned. For instance, the lower layers may remain frozen, while the top layers are adapted to the specific task [326]. This method is useful when labeled data is limited, as it reduces the number of parameters that need to be updated.

- **Head Fine-Tuning:** In this approach, the pretrained model is used as a fixed feature extractor, and only the final classification layer is trained on the task-specific data [215]. This is often employed when computational resources are limited, as it reduces the number of parameters being updated.

2.1.5 The Role of Parametric Knowledge in Question Answering

In recent years, LLMs have demonstrated remarkable performance in QA tasks [28, 207]. One key factor contributing to their success is the ability to encode vast amounts of factual knowledge directly within their parameters [219]. This ability allows these models to generate accurate answers to a wide variety of questions without needing explicit access to external knowledge sources. By leveraging the rich internal representations learned during pretraining, LMs can tackle a broad range of QA tasks with little to no task-specific fine-tuning [28].

Limitations and Gaps in Parametric Knowledge: While the parametric knowledge embedded in LMs is vast and useful, it is not without limitations. The model's knowledge is inherently bound by the data it was trained on, which may include biases, inaccuracies, or outdated information [260]. For example, if a model was trained before a significant event, such as the election of a new president or the discovery of a scientific breakthrough, it may provide an outdated answer when queried about these topics [260]. Additionally, because the knowledge is encoded in the model's parameters, it is not always directly interpretable, and the model may not be able to explain how it arrived at a particular answer.

Moreover, while LMs excel at factual recall, they can sometimes struggle with tasks that require reasoning, such as answering questions that involve multiple-step logic or complex relationships [325]. For example, a question like "How does the process of photosynthesis contribute to the oxygen cycle?" requires understanding both the biological process and its role in a broader ecological system. While LLMs may generate an answer, they may not always capture the full depth of such multi-step reasoning or might produce a response that lacks the nuance required to accurately address the question.

2.1.6 Improving Parametric Knowledge Utilization

While parametric knowledge encoded in large language models is highly valuable, there are techniques that enhance the model's ability to utilize this knowledge more effectively. Some of the key approaches include:

- Chain of Thought (CoT)
- Prompting (In-context learning)

2.1.6.1 CoT

CoT prompting [305] is a technique that enhances a model's ability to reason through complex problems by encouraging it to break down tasks into a sequence of intermediate steps. Instead of simply providing a final answer, CoT involves prompting the model to generate a step-by-step reasoning process that leads to the final conclusion.

This approach mimics human reasoning, where complex problems are often solved through a series of smaller, more manageable steps. By explicitly prompting the model to generate intermediate reasoning steps, CoT improves the model's ability to tackle questions that require logical deductions, multi-step reasoning, or deep understanding.

CoT has been particularly effective in tasks that involve arithmetic reasoning, commonsense reasoning, and more nuanced question answering.

2.1.6.2 In-context Learning

In-context learning [60] refers to the practice of providing a model with examples or context within the prompt itself to guide the model's behavior and improve task performance. In this approach, the model is given a few examples of the desired task or a specific context within the input, and it learns to adapt its responses accordingly.

Unlike traditional training methods, where a model is explicitly fine-tuned on labeled data, prompting allows the model to learn how to perform a task in a more flexible and dynamic way. By presenting the model with examples of input-output pairs within the prompt, the model is able to use the context to infer how it should respond to new, unseen inputs.

In-context learning is particularly useful for zero-shot or few-shot learning scenarios, where the model is tasked with performing a new task with little to no task-specific fine-tuning. By providing the model with a rich context in the prompt, prompting can guide the model to generate more accurate and relevant responses for a wide range of tasks, from text generation to question answering.

2.1.7 Integration with External Knowledge

Although parametric knowledge alone is powerful, there is increasing interest in combining LMs with external knowledge bases to create hybrid systems that offer the best of both worlds. By integrating a model's internal knowledge with external sources such as knowledge graphs, encyclopedias, or databases, it is possible to build more robust question answering systems. This integration can enhance the model's performance by providing access to more detailed, domain-specific, or up-to-date information that the model may not have learned during pretraining.

One prominent example of this integration is Retrieval-Augmented Generation (RAG) [83]. In RAG, the model retrieves relevant external information, such as documents, KG triples, or other relevant data, from an external knowledge base during inference. This retrieved information is then used as additional context to help the model generate more accurate and grounded responses.

The key advantage of RAG is its ability to combine parametric knowledge (learned during pretraining) with external sources of information, allowing the model to access both static knowledge and the latest, most specific facts available. This makes RAG especially useful in domains where knowledge is constantly evolving or is highly specialized, such as scientific research, medical, or legal contexts, where models require access to real-time or verified data to provide reliable answers.

2.2 Question Answering Systems

2.2.1 Definition and Significance of QA Systems

A QA system is a computational tool designed to understand and respond to queries posed by users in natural language. These systems vary in complexity, from simple keyword-based search engines to advanced AI-driven models capable of comprehending context and providing nuanced responses [207]. QA systems can operate across diverse domains, such as medical [327, 328, 314], and general knowledge [207], making them valuable tools for efficiently accessing and understanding information.

One of the features of QA systems is their adaptability across diverse domains. Like many NLP systems, they are not confined to a single domain but rather extend their utility across various fields, including but not limited to medical, general knowledge, technical support, customer service, and legal documentation. This versatility underscores their significance as invaluable tools for efficiently accessing and comprehending information across different disciplines.

By leveraging vast repositories of medical literature, databases, and clinical records, these systems can empower clinicians to make informed decisions and deliver optimal patient care. Notable advancements in medical QA systems include models like LinkBERT [327], QAGNN [328], and MedConQA [314], which demonstrate the efficacy of QA systems in extracting relevant medical information and providing contextually appropriate answers to complex medical queries.

In the realm of general knowledge, QA systems serve as indispensable tools for information retrieval and synthesis. Whether it is seeking historical facts, scientific principles, geographical

data, or cultural insights, users can pose questions in natural language, and the QA system sifts through vast repositories of structured or unstructured data to provide a concise, accurate responses. Notable examples of AI-driven QA systems include those developed by OpenAI [207], which leverage advanced language models like GPT to comprehend context, infer meaning, and generate human-like responses to many queries.

Many QA systems use a blend of techniques from NLP, machine learning, information retrieval, and knowledge representation. These systems typically entail processes such as natural language understanding, entity recognition, semantic parsing, information retrieval, and answer generation. NLU techniques enable QA systems to decipher the syntactic and semantic structure of user queries, identifying key entities, relations, and intents. Entity recognition [208] plays a critical role in identifying relevant entities mentioned in the query, such as names of people, organizations, locations, and temporal expressions. Semantic parsing involves mapping natural language queries to structured representations, facilitating efficient retrieval of relevant information from knowledge bases and corpora. Information retrieval techniques enable QA systems to efficiently search through vast repositories of textual data or other forms, such as graphs, retrieving documents or passages containing potentially relevant information. Finally, answer generation involves synthesizing the retrieved information into coherent, grammatical, contextually appropriate responses tailored to the users' queries.

2.2.2 Types of QA Systems

QA systems can be categorized according to their output mechanisms. For instance, extractive QA systems [164], extract answers directly from a given context, often by predicting start and end token indices. Conversely, multiple-choice (MC) QA systems [254], provide a probability distribution across answer choices, with users typically selecting the answer with the highest probability. In contrast, generative QA systems [290], produce answers in free-form text.

Furthermore, QA systems can also be classified based on their input modalities, which may include structured inputs like tables [198], knowledge graphs [249], and unstructured text passages

[138].

While commonly generative QA systems that leverage external knowledge are referred to as open-domain QA systems [352, 339], it is noteworthy that external knowledge integration is feasible across various QA system types. Furthermore, in recent years such systems have also been termed retrieval-augmented systems [147, 111, 263] and knowledge-enhanced systems [167, 11, 303, 161, 63, 55]. In this thesis, our focus is not on the retrieval system and hence our focus is solely on knowledge-enhanced systems.

2.2.3 Limitations of QA Systems

Despite their significant importance, QA systems encounter numerous obstacles that hinder their effectiveness. These systems, which have gained attention for their ability to understand and respond to natural language queries, represent a major shift in various fields like customer service, education, and healthcare. However, beneath their impressive abilities lie challenges that affect their performance and reliability.

In the next subchapters, I will discuss some of the difficulties QA systems face, highlighting the complexities of developing AI models that can understand and generate accurate responses to human questions. From dealing with ambiguous language to understanding context, these challenges show the difficulty of creating QA systems that match human intelligence.

Furthermore, QA systems rely heavily on extensive data repositories, adding another layer of complexity. Incomplete or noisy datasets can distort learning, leading to inaccuracies and biases in responses. Additionally, the ever-changing nature of language and information requires constant adaptation and refinement of these systems to stay relevant and effective.

Despite these challenges, efforts to improve QA systems continue. By tackling these obstacles and leveraging advancements in ML and NLP, researchers aim to overcome the limitations of current QA systems and unleash their full potential in shaping the future of AI-driven communication and question answering.

2.2.3.1 Biases

One significant challenge QA systems face pertains to bias, wherein the responses generated may inadvertently reflect societal or cultural prejudices embedded within the data utilized for their training [255].

The complexity of societal dynamics and their impact on health outcomes is highlighted by the World Health Organization, emphasizing the influential role of social determinants such as racism, sexism, and discrimination. These factors are often deemed more critical than traditional healthcare interventions or lifestyle choices in shaping health outcomes. Consequently, it becomes imperative for biomedical NLP systems to remain impervious to influences unrelated to biological or medical factors, ensuring equitable treatment of users irrespective of extraneous attributes such as names. The prevalence of social biases across diverse NLP training datasets and models is well-documented, ranging from gender biases observed in machine translation systems to racial biases evident in predictions related to opioid misuse.

Extensive research is devoted to investigating social biases within QA systems, with studies employing demographically modifications of names, gender, and others, to uncover underlying biases [144]. Furthermore, a multitude of investigations into bias within natural language generation systems, transformers, and analogous models have shed light on outputs influenced by various demographic factors [255].

Within the medical domain, there is a concerted effort to evaluate biases inherent in AI models [255]. For instance, previous research have analyzed biases related to unhealthy alcohol use risk among classifiers utilizing electronic health records in trauma patients [25]. Similarly, examinations of gender and ethnicity biases in pain management settings have been conducted [162], contrasting the performances of different generations of AI models. However, obtaining unbiased datasets for investigating model biases poses considerable challenges, particularly in domains like pain management where societal biases are deeply ingrained. Consequently, alternative data sources are often leveraged to facilitate unbiased analyses. Moreover, studies evaluating racial biases within

clinical settings further underscore the importance of addressing bias within AI applications [255]. Additionally, efforts are being made to utilize NLP systems to evaluate whether language patterns indicative of bias or stereotypes are present within medical exams, highlighting the importance of bias mitigation efforts within NLP [255].

2.2.3.2 Hallucinations

Despite their impressive performance across various benchmark tasks, QA systems still struggle with producing incorrect or hallucinatory content [237, 252, 151, 66]. Despite appearing confident, these systems may provide factually inaccurate statements. This discrepancy between confidence and accuracy presents challenges for applications where output correctness is crucial, indicating that such models might not be ideal for scenarios requiring precision and reliability, such as in medicine. This issue is not consistent across all models, but tends to be more common in those trained on languages with limited resources.

In the realm of low-resource LMs, challenges related to data scarcity and insufficient curation are especially important. These models often operate with limited resources, making them more prone to generating inaccurate or misleading responses. The lack of high-quality training data worsens the situation by restricting the model's exposure to varied and dependable information sources. As a result, hallucinations by the model in such contexts can have harmful patterns ingrained in the training data, further compromising the reliability of the generated content.

The issue of factuality within LMs is complex and goes beyond mere data scarcity. While LMs store vast amounts of factual knowledge and act as repositories of information, the accuracy of the retrieved information is not guaranteed. Despite their capability to retrieve relevant information, these models may struggle to differentiate between factual and non-factual content.

The challenge of accuracy is apparent across various applications, including QA, dialogue systems, image captioning, text summarization, and translation. In each domain, the accuracy of the generated content relies on the model's ability to access and process factual information reliably. However, the lack of transparency regarding the origins of the model's knowledge complicates

efforts to verify its factuality. Without clear insights into the sources of information (i.e., citation), distinguishing between factual and non-factual content becomes challenging.

Addressing the factuality issue in LMs necessitates a comprehensive approach that accounts for both technical and ethical considerations. Technically, improving the quality and diversity of training data can help reduce the risk of generating inaccurate or misleading content. Additionally, incorporating mechanisms for fact-checking and validation into the model architecture can enhance users' confidence in the accuracy of the generated responses. Ethically, promoting transparency and accountability in the development and deployment of LMs is crucial for ensuring responsible use and mitigating the spread of misinformation.

In conclusion, while LMs offer significant potential for accessing and disseminating information, their tendency to generate inaccurate or misleading content presents significant challenges. Addressing the accuracy issue requires concerted efforts from researchers, developers, and policymakers to establish robust mechanisms for verifying the accuracy of generated content. By confronting this challenge proactively, we can leverage the full potential of LMs while guarding against the spread of misinformation.

2.2.3.3 Long Contexts

QA systems also frequently encounter challenges in grasping lengthy contexts, particularly when the relevant information is found within the middle sections. This issue poses a significant hurdle to their performance, as highlighted by [157], who coined it as the "lost in middle" problem. Essentially, as the context extends, models may lose track of the crucial details found within the text. This loss of focus severely impacts their ability to generate accurate and coherent responses. Consequently, addressing this limitation is pivotal for enhancing the effectiveness of LMs across various tasks, from natural language understanding to generation.

2.2.3.4 Robustness

Recent research has focused heavily on issues like the robustness of QA systems when faced with adversarial attacks [152, 259]. One major concern is how susceptible these systems are to context-based interference, where even seemingly irrelevant distractions can significantly impact their performance. This susceptibility has led to numerous investigations into different methods to strengthen QA models against such challenges.

Previous research introduced a technique involving the insertion of sentences resembling questions or random distractor words [152], resulting in a notable performance drop of over 50. However, some argue against this method [330], suggesting that models could easily identify such artificial distractors and disregard them and proposed a modification involving the repositioning of distractors while adding extra fake answers to improve resilience. Subsequent studies, experimented with further adjustments, such as shuffling distractors [273].

More research has explored various methods to create adversaries for QA systems [169]. Some efforts introduced new techniques, resulting in significant performance drops. Others focused on manipulating context or introducing noise [259], both of which have been shown to decrease model performance.

These discussions underscore two main points: firstly, models can be easily influenced by a wide range of distractions, even those lacking semantic coherence. Secondly, the type and complexity of the distractor significantly affect the extent of performance degradation.

To tackle these challenges, researchers have explored various strategies. One common approach involves training models with augmented noisy data, as demonstrated by several studies [239, 330]. However, some studies caution that this approach may have limited benefits. Alternatively, efforts have been made to train models to edit distractor information, or prompt systems to ignore irrelevant information [18]. These diverse strategies reflect ongoing efforts to enhance the robustness of QA systems against contextual noise and adversarial attacks.

2.2.3.5 Ambiguous Contexts

The ongoing endeavor to enhance QA systems, particularly in navigating ambiguous inquiries, remains a central focus, necessitating continuous research and development efforts to improve their efficacy and reliability. One significant area of research involves investigating how these systems handle contexts that conflicts with their parametric knowledge. Conflicting contexts may interfere with a model's existing knowledge in familiar scenarios, but they might not always arise in contexts involving unknown knowledge splits, as the alternative context could align with the model's existing knowledge.

To assess how systems respond to conflicting knowledge, various methodologies have emerged, with entity substitution being a prominent approach [163]. One approach explores substituting original answer entities with similar alternatives, shedding light on factors influencing models' reliance on parametric knowledge, while another focus on enhancing systems' robustness to conflicting knowledge through prompts [259]. Others investigate the impact of information retrieval systems on a model's use of parametric knowledge, or delve into disentangling a system's parametric and contextual knowledge [199], revealing vulnerabilities across different learning settings and observing decreases in performance when confronted with conflicting entities.

Another prevalent approach involves utilizing negations. One research tailor contexts to each Transformer model, showing their sensitivity to negation [90]. However, models often persist in predicting the original answer despite negations. Additionally, masked LMs are used to introduce conflicting knowledge, yielding varying observations regarding vulnerability and contextual coherence [210, 152].

As the landscape of knowledge changes, strategies for adapting QA systems continue to progress. Proposed methods range from modifying factual knowledge within Transformer models to employing hyper-networks for predicting system weight updates [52]. Some utilize auxiliary networks to refine pretrained model behavior or identify and update weights relevant to factual information [189]. Despite proposed misinformation detection mechanisms and strategies to enhance

contextual coherence, challenges in generalization persist. Furthermore, advocating for carefully designed prompting strategies, the generation of both parametric and contextual answers, and exploring avenues like external storage of edited facts or entity-based masking are suggested for refining QA systems amidst evolving challenges.

2.2.4 External Knowledge and Open-Domain QA Systems

Many modern QA systems integrate external knowledge to improve their performance, especially in open-domain settings. Open-domain systems [352, 339] leverage retrieval-based methods to incorporate external data sources, enabling them to answer questions outside the scope of their training data. This class of systems is sometimes referred to as *retrieval-augmented* [147, 111, 263] or *knowledge-enhanced* [167, 11, 303] systems. These systems access and integrate structured or unstructured external knowledge to provide more accurate and contextually appropriate responses, making them valuable in dynamic environments where new or domain-specific information is frequently updated.

In this thesis, the focus is primarily on knowledge-enhanced QA systems, rather than the retrieval component itself.

2.3 Knowledge Integration in Question Answering

QA systems that utilize external knowledge sources must integrate this external information with the internal model knowledge in a coherent and efficient manner. The integration of external knowledge into QA systems can significantly enhance the quality of answers, especially for complex and domain-specific queries. This section discusses various sources of external knowledge, retrieval techniques, integration methods, and the challenges encountered during this process.

2.3.1 Sources of External Knowledge

External knowledge used in QA systems can be broadly categorized into *structured* and *unstructured* sources. These sources vary in their organization and accessibility, but both types can

provide valuable information for answering questions.

- **Structured Knowledge:** These are highly organized sources that store information in a predefined schema, such as KGs and databases. KGs are often used to represent entities and their relationships, providing rich contextual information that can be leveraged by QA systems. Databases, including relational and NoSQL databases, also offer structured data that can be queried for specific facts, making them useful for retrieving precise, factual answers.

- **Unstructured Knowledge:** These sources include documents, web content, and large-scale corpora such as Wikipedia, scientific literature, or online articles. Unstructured data is more flexible and can contain a wide range of information, but it requires effective techniques for extracting relevant content. The open-ended nature of unstructured knowledge makes it challenging to retrieve exactly the information needed for a particular question.

2.3.2 Techniques for Knowledge Retrieval

Once external knowledge sources are identified, the next step is retrieval, where relevant information is fetched from these sources to answer the user's question. Retrieval can be achieved through various techniques, two of the most prominent being *classical retrieval methods* and *embedding-based retrieval*.

- **Classical Retrieval Methods:** These methods rely on traditional information retrieval techniques such as *BM25*, a probabilistic model that ranks documents based on the presence of query terms [12]. Classical retrieval methods work by matching keywords in the query to those in the documents and retrieving the most relevant results. These techniques are fast and effective, particularly in domains where the answer can be directly found in structured or well-indexed documents.

- **Embedding-Based Retrieval:** This approach uses neural network-based models to map both the question and candidate knowledge sources (such as passages) into dense vector

representations (embeddings). Models like BERT, RoBERTa, or DPR [124] generate these embeddings, capturing the semantic meaning of both the question and the documents. The most relevant documents are retrieved by comparing the question's embedding with those of the candidate passages, typically using similarity metrics like cosine similarity. Embedding-based retrieval is effective in capturing the contextual relationships between words and sentences, which helps retrieve more semantically relevant information even when the exact phrasing doesn't match the query.

2.3.3 Techniques for Knowledge Integration

Once relevant information is retrieved, the next challenge is to integrate this external knowledge with the internal representations of the model. Several techniques have been proposed to achieve effective integration.

- **Concatenation of Context and Question:** One straightforward approach is to concatenate the external knowledge (context) with the question into a single sequence [28]. This technique treats the combination as a unified input for the model, where the model learns to process both the question and the context simultaneously.

- **Encoding and Aggregation:** Another technique involves separately encoding the question and the retrieved knowledge, followed by aggregating the encoded representations. This can be achieved through attention mechanisms or other fusion techniques [223]. The aggregated representations are then used to generate an answer. This method allows the model to weigh the importance of different knowledge sources independently before combining them.

2.3.4 Challenges in Combining Internal and External Knowledge

While integrating external knowledge into QA systems can greatly improve performance, several challenges must be addressed, including fact-checking, consistency, and hallucinations.

- **Fact-Checking:** The integration of external knowledge introduces the challenge of verifying

the accuracy and reliability of the retrieved information [257]. Ensuring that only factually correct information is used in the answer is a crucial challenge in knowledge integration.

- **Consistency:** External knowledge must be integrated in a manner that maintains consistency with the internal model's understanding. If conflicting information from external sources is integrated without careful handling, it can lead to contradictions in the final answer [260]. Ensuring that the model can resolve conflicting facts or at least acknowledge uncertainty is an ongoing challenge.

- **Hallucinations:** Hallucinations refer to the generation of incorrect or fabricated information that is not grounded in the retrieved knowledge [237, 252, 151, 66]. This is particularly problematic in generative models that rely heavily on external knowledge. Hallucinations can result from poor retrieval or flawed integration strategies, and developing methods to mitigate this issue is critical for improving the reliability of QA systems.

In conclusion, knowledge integration is a key component of modern QA systems, enabling them to answer questions by utilizing both internal and external knowledge sources. While effective techniques for retrieval and integration have been proposed, challenges such as fact-checking, consistency, and hallucinations remain central to the development of robust and reliable QA systems.

2.4 Dynamic Environments and Their Challenges

In the context of this thesis, dynamic environments refer to settings where the available knowledge changes over time. This section defines dynamic environments, discusses the challenges posed by evolving knowledge, and explores strategies for handling these challenges in QA systems.

2.4.1 Definition of Dynamic Environments in QA

Dynamic environments in QA are characterized by the continuous evolution of knowledge. Knowledge in such environments is not static; it may change due to new discoveries, updates, or shifts in consensus. For example, a dynamic environment could involve updates in political positions,

scientific breakthroughs, or the reevaluation of previously established facts. Such changes may stem from:

- **Shifts in Factual Knowledge:** For example, political leadership changes (e.g., a new president), or research findings that render prior conclusions outdated or false.
- **Conflicting Information:** New information may contradict previously established knowledge, leading to conflicts between sources. This could involve a situation where new evidence challenges earlier research, or when subjective opinions or biases influence the portrayal of facts.
- **Subjective Information and Opinions:** In some cases, the environment may not have a definitive factual answer. Opinions or subjective viewpoints can evolve, making it challenging for a QA system to distinguish between fact and opinion or decide which viewpoint is most relevant.

In such dynamic environments, QA systems must be able to incorporate these changes and adapt their responses accordingly to maintain accuracy and relevance.

2.4.2 Handling Evolving Knowledge

Handling evolving knowledge in dynamic environments presents significant challenges for QA systems, requiring both effective retrieval and continuous adaptation. Several strategies can be employed to address these challenges:

- **Leveraging External Knowledge:** By using external knowledge sources, such as up-to-date databases, real-time news feeds, and online resources like Wikipedia, QA systems can access the latest information. External knowledge retrieval techniques, such as embedding-based retrieval or RAG, can help in identifying new, relevant facts or resolve contradictions between different sources.

- **Generating Multiple Answers for Conflicting Contexts:** In scenarios where conflicting information arises, a robust approach is to generate multiple possible answers based on the different sources of knowledge [260]. This strategy involves generating answers that acknowledge the ambiguity or conflict within the context, thus providing a more comprehensive response. A QA system can include confidence scores or uncertainty indicators to inform users when multiple perspectives are possible.

- **Retraining the Model:** A more long-term solution to handle evolving knowledge is to retrain the QA model periodically with updated data. This ensures that the model remains aware of the latest developments and knowledge shifts. Retraining involves fine-tuning the model on new datasets, incorporating recent facts, and resolving conflicts based on the latest information. It also enables the system to recalibrate its understanding when new knowledge contradicts its previously learned patterns.

By combining these techniques, QA systems can remain adaptable and capable of responding to the dynamic nature of knowledge in real-world environments.

Chapter 3

Knowledge Probing

Correctly answering questions often requires leveraging both external knowledge sources (e.g., databases, documents, KGs) and internal, parametric knowledge stored within model parameters. Since parametric knowledge plays a significant role in shaping model outputs, a substantial portion of my research focuses on probing and assessing what models learn internally, including both text-based and KG-grounded systems.

Knowledge probing entails investigating the stored knowledge within a LM [219]. This method involves examining the model with various prompts or questions designed to gauge its comprehension or retrieval of factual information.

One typical method of knowledge probing involves creating tasks that test different facets of the model's knowledge. These tasks can vary in complexity, ranging from straightforward factual inquiries to more intricate reasoning exercises. For instance, a probing task may entail asking the model to identify the capital of a given country [219].

Researchers employ knowledge probing to assess the strengths and limitations of LMs and to evaluate their applicability for particular tasks [259, 256]. Through analyzing the model's responses to probing tasks, researchers can gain insights into its understanding of various topics, its capacity for logical reasoning, and its reliance on external knowledge sources.

Furthermore, knowledge probing serves to pinpoint biases or deficiencies in a LMs comprehension. For instance, if a model consistently provides inaccurate answers to questions about specific subjects [255], it may indicate a lack of relevant knowledge or skewed representations.

Moreover, knowledge probing aids in refining language models by identifying areas where they perform sub-optimally. By pinpointing weaknesses, researchers can focus on enhancing the model's performance in those areas through targeted training or fine-tuning.

In summary, knowledge probing serves as a valuable tool for comprehending and evaluating LMs, as well as for advancing the field of NLP. By systematically examining a model's knowledge and reasoning abilities, researchers can gain deeper insights into its capabilities and limitations, ultimately leading to the development of more robust and effective AI systems.

3.1 Emerging Challenges in Personalized Medicine: Assessing Demographic Effects on Biomedical Question Answering Systems

While biases in QA models are well-documented, their implications in sensitive domains like biomedicine remain underexplored. **The work described in this section has been published in AACL 2023 [255]**.

3.1.1 Introduction

Natural language processing (NLP) has long been used in health care and life sciences. However, NLP systems exhibit surprising behaviors that can be difficult to predict or control: problems with general-purpose NLP systems reflecting stereotyping and stigmatizing biases have been apparent since the Microsoft Taybot debacle in 2016 and remain a major issue to this day [197, 245, 23, 248, 334].

The World Health Organization states that social determinants of health, including the experience of racism, sexism, and other forms of discrimination, "can be more important than health care or lifestyle choices in influencing health."[1] Thus, for biomedical NLP systems it is of particular importance to not be affected by factors irrelevant to biology and medicine, and for researchers to ensure they serve their users fairly irrespective of irrelevant attributes, such as names, as shown in Figure 5.2. Here, we test the effect irrelevant demographic information has on biomedical QA

[1] https://www.who.int/health-topics/social-determinants-of-health#tab=tab_1

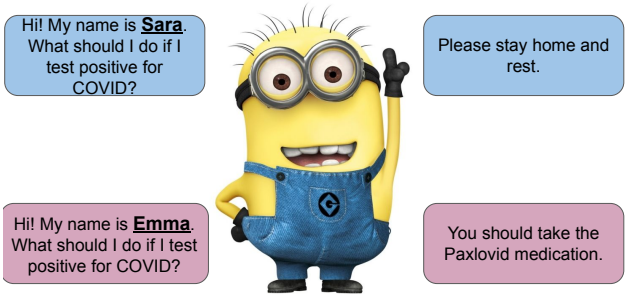


Figure 3.1: An undesired behavior from a biomedical QA system: the model changes its answers when provided with different biomedically irrelevant information (e.g., names).

systems. As a test-bed, we choose a subset of questions from the US Medical Licensing Exam level 1 [USMLE1; 115] **whose answers, according to two medical professionals, are independent of the patient's demographics.** Although the questions are multiple-choice, correct answers require broad medical knowledge, including diagnosis and treatment of all common diseases, as well as an understanding of the underlying molecular and physiological mechanisms, potential drug side effects, probabilistic reasoning, and more.

We add irrelevant demographic information in a controlled way to the USMLE questions in order to answer the following research questions: (RQ1) Do the models' answers change when being provided with irrelevant demographic information? (RQ2) Is the answer to RQ1 different for KG-grounded and text-based QA systems? We experiment with two biomedical QA systems: BioLinkBERT [327], a text-based model, and QAGNN [328], which is the highest performing KG-based model on USMLE.

There are good reasons to believe that neither system should be affected by irrelevant patient information: both are trained solely on biomedical text, which is most often independent of irrelevant demographic information, and QAGNN is additionally grounded by a KG that does not contain any demographic representations. Unfortunately, we find that both systems change many of their answers when provided with irrelevant patient demographic information. We also observe that the two systems differ in which demographic information affects them. Finally, we compare biomedical

Dimensionless	A 23-year-old **patient** presents to a psychiatrist for evaluation of situational anxiety. The **patient** reports that **they** recently started a new job and is very stressed.
Ethnicity	A 23-year-old **Black patient** presents to a psychiatrist for evaluation of situational anxiety. The **patient** reports that **they** recently started a new job and is very stressed.
Gender	A 23-year-old **female** presents to a psychiatrist for evaluation of situational anxiety. **She** reports that **she** recently started a new job and is very stressed.
Names	A 23-year-old **patient named Tom** presents to a psychiatrist for evaluation of situational anxiety. The **patient** reports that **they** recently started a new job and is very stressed.
SOr	A 23-year-old **bisexual patient** presents to a psychiatrist for evaluation of situational anxiety. The **patient** reports that **they** recently started a new job and is very stressed.
SOr+Gender	A 23-year-old **bisexual female** presents to a psychiatrist for evaluation of situational anxiety. **She** reports that **she** recently started a new job and is very stressed.
Ethnicity+Gender	A 23-year-old **Asian male** presents to a psychiatrist for evaluation of situational anxiety. **He** reports that **he** recently started a new job and is very stressed.
Ethnicity+Gender +Names	A 23-year-old **Hispanic female named Guadalupe** presents to a psychiatrist for evaluation of situational anxiety. **She** reports that **she** recently started a new job and is very stressed.

Table 3.1: Dimensions example. Given a question, for each dimension, we demographically-enhance the question by adding relevant words (e.g., Black, bisexual, named X) and changing its gender tokens in order to create multiple datasets for the specific dimension. SOr=sexual orientation.

to generic systems (i.e., trained on generic English text) and find that, as expected, the generic system changes even more of its answers in most cases (up to 17% for gender). However, for some demographics, such as sexual orientation, the biomedical system changes up to 23% of its answers. We hope that shedding light on this problematic behavior will motivate future work to further investigate its impact as well as possible solutions.

3.1.2 Experimental Setup

3.1.2.1 Motivation

Biomedical QA systems can be beneficial for both healthcare providers and patients for many reasons: 1) With traditional search engines, finding reliable medical information can take time and effort due to the vast amount of unfiltered content available online, while QA systems allow users to quickly find answers; 2) such systems can serve as powerful learning tools for students and residents seeking to deepen their understanding of complex medical topics; 3) in low-resource settings there may be limited access to qualified healthcare professionals, which leads to delayed or incorrect diagnoses that may worsen health outcomes over time. Fortunately, biomedical QA systems can bridge this gap and extend the reach of health services to vulnerable populations worldwide.

However, in order for such systems to be safely deployed, ensuring that they provide fair behavior towards patients is critical. For example, imagine that a White and an African-American

patient present themselves with similar symptoms at a hospital and that none of their symptoms indicate a problem related to their ethnicity. If one was treated with the correct medication while the other received an incorrect one, this would be highly problematic. Thus, it is important to understand if current biomedical QA systems could result in such an outcome.

3.1.2.2 MedQA-USMLE

The MedQA-USMLE dataset [115] is an open-domain QA dataset, which covers three languages: English, traditional Chinese, and simplified Chinese. MedQA has medical questions which represent real-world scenarios and evaluate physicians on their clinical decision making skills. The questions are varied and require a significant understanding of medical concepts. Here, we choose to only use the English version, which is composed of 12,723 multiple-choice prompts taken from the professional medical board exams. Each prompt consists of *context* and *question*, e.g., *"An 18-year-old male presents to the emergency room smelling quite heavily of alcohol and is unconscious. A blood test reveals severe hypoglycemic and ketoacidemia. A previous medical history states that he does not have diabetes. The metabolism of ethanol in this patient's hepatocytes resulted in an increase of the [NADH]/[NAD+] ratio. Which of the following reaction is favored under this condition?"*. Each question comes with four answer choices. The options for the above example are: *Pyruvate to acetyl-CoA, Citrate to isocitrate, Oxaloacetate to malate*, and *Oxaloacetate to phosphoenolpyruvate*.

3.1.2.3 Question Selection

Some phenomena are more prevalent in certain populations, such as pregnancy [278] or prostate cancer. For other diagnoses, patient demographic information is irrelevant and should accordingly not be taken into account. For our experiments we build a dataset consisting of **only questions whose answers do not depend on sex, ethnicity, or sexual orientation.** We do so by following [162]'s approach and extract 100 vignettes, which are designed to allow for the inclusion of diverse ethnics and gender "profiles" in order to assess potential biases. These vignettes

	Random	Gender		Ethnicity					SOr			Gender+Ethnicity									Gender+SOr					
		M	F	W	A-A	B	H	As	Hetero	Bi	Homo	M+W	M+A-A	M+B	M+H	M+As	F+W	F+A-A	F+B	F+H	F+As	M+Hetero	M+Bi	M+Homo	F+Hetero	F+Bi	F+Homo
QAGNN	2	6	**7**	6	**9**	7	6	6	**9**	**7**	**15**	**8**	**8**	**9**	**10**	**8**	**9**	**10**	9	**9**	**9**	8	6	11	**10**	8	9
BioLinkBert	2	6	6	6	8	7	**7**	**11**	6	6	14	6	7	5	7	6	8	8	9	8	8	**9**	**9**	**23**	8	**13**	**23**

Table 3.2: Percentage of questions with changed answers as compared to a question with no demographic information about the patient. *M*=male; *F*=female; *W*=White; *B*=Black; *A-A*=African-American; *H*=Hispanic; *As*=Asian; *SOr*=sexual orientation; Random=Random change as described in Section 3.1.3.

are verified by two medical experts to be demographics-independent, and after the demographics-enhancing process, which will be discussed in the next section, result in **16,700 questions overall**, which are used to evaluate the effect irrelevant demographic information has on QA systems.

3.1.2.4 Demographics-enhanced Dataset Creation

We experiment with the following types of modified questions: dimensionless (i.e., no demographic information), ethnicity, gender, names, sexual orientation, gender+ethnicity, gender+sexual orientation, and gender+ethnicity+names.

The reasoning for each chosen dimension are as follows: dimensionless shows no demographic information, and hence will be used as a baseline to compare how many of the answers change when we add irrelevant demographic information. Ethnicity, sexual orientation, and gender, while not always shown in medical text, are sometimes mentioned when the demographic information is relevant. Hence, we want to see if the models associate any medical conditions with them. We use two genders, but expect that our results will generalize to additional genders. As for names, these are clearly not medically relevant ever and are rarely shown in medical text. Hence, we choose them to see if there are unexpected differences in answers change.

Ethnicities include White, Black, African-American, Hispanic, and Asian. Genders include male and female. Names include the 10 names for each ethnicity from the Q-Pain dataset, which

		Gender		Ethnicity					SOr		
		M	**F**	**W**	**A-A**	**B**	**H**	**As**	**Hetero**	**Bi**	**Homo**
Correct → Incorrect	QAGNN	1	**3**	**4**	**4**	**4**	3	3	**4**	**3**	**6**
	BioLinkBert	1	2	2	3	3	3	**6**	2	1	5
Incorrect→Incorrect	QAGNN	2	1	2	2	**3**	**3**	3	**3**	3	6
	BioLinkBert	**4**	**2**	2	**4**	2	2	3	2	3	6
Incorrect→Correct	QAGNN	**3**	**3**	0	**3**	0	0	0	2	1	3
	BioLinkBert	1	2	**2**	1	**2**	**2**	**2**	2	**2**	3

Table 3.3: Percentage of answers that changed from from correct to incorrect, incorrect to incorrect, and incorrect to correct for each model. *M*=male; *F*=female; *W*=White; *B*=Black; *A-A*=African-American; *H*=Hispanic; *As*=Asian; *SOr*=sexual orientation.

originated from the Harvard Dataverse's *Demographic aspects of first names* dataset [292]. And while "Black" and "African American" are largely synonymous, we want to see if they are different from the models' perspective. Notably, to medically-untrained users, all of these may seem relevant and hence potentially be added to queries when such users request medical assistance.

We follow a similar process as the creators of the Q-Pain dataset and make each context, question, and answer (CQA) as neutral as possible. Given a CQA, such as "A 23-year-old female presents to a psychiatrist...", we first automatically mask any word that indicates gender (e.g., male, female, he, she, wife, boyfriend): "A 23-year-old [GENDER_MASK] presents to a psychiatrist...". Then, given a dimension (e.g., gender), we automatically replace each unique masking with their corresponding token replacement (e.g., replacing "[GENDER_MASK]" with "male").

Overall, each of these dimensions and their variations augment each of the 100 vignettes and result in overall 16,700 questions. See Table 3.1 for examples. And while we only use the English version of the dataset, this process can be easily applied to other languages. The data will be publicly available and have an MIT License.

3.1.3 Random Change

We use a version of the questions with no demographic information, and, in each prompt's first sentence, replace the word "patient" with "person". With this we examine the effect of a small

	O*	**O**	**D**	**Gen.**		**Ethnicity**					**SOr**			**Gender+Ethnicity**									**Gender+SOr**					
				M	**F**	**W**	**A-A**	**B**	**H**	**As**	**Hetero**	**Bi**	**Homo**	M+W	**M+A-A**	**M+B**	**M+H**	**M+As**	**F+W**	**F+A-A**	**F+B**	**F+H**	**F+As**	**M+Hetero**	**M+Bi**	**M+Homo**	**F+Hetero**	**F+Bi**	**F+Homo**
1	*38*	**40**	40	**42**	40	36	**39**	36	37	**37**	38	38	37	38	**38**	36	35	37	35	34	35	35	35	36	39	**38**	36	36	**37**
2	***40***	39	40	40	40	**40**	38	**39**	**39**	36	**40**	**41**	**38**	**39**	37	**41**	**40**	**40**	**40**	**36**	**40**	**41**	**40**	**41**	**41**	36	**41**	**40**	36

Table 3.4: Accuracy (in percentages) of the two models on our demographically enhanced datasets. *M*=male; *F*=female; *W*=White; *B*=Black; *A-A*=African-American; *H*=Hispanic; *As*=Asian; *SOr*=sexual orientation; O*=original test dataset; O=the original, unmodified 100 vignettes; D=No demographic information; Gen=Gender; 1=QAGNN; 2=BioLinkBERT.

but insignificant textual variation on each model. We choose this change over others (e.g., adding random words, irrelevant demographics, or fictitious cities) as this reduces the possibility of models changing their answers due to the context such random words had in the training data (e.g., Africa is more prevalent to the sleeping sickness disease than the US). Moreover, neither "person" nor "patient" reveal information about the human.

3.1.4 Models

We compare two existing algorithms: QAGNN [328] and BioLinkBert [327]. While better models exist for the USMLE dataset, many of them have billions of parameters and we are unable to test them for computational reasons. That being said, BioLinkBert is currently among the state of the art on the USMLE dataset, and QAGNN is the top (and, to the best of our knowledge, only) KG-grounded model. We use existing implementations and models and describe both systems in the following.

3.1.4.1 QAGNN

The main component of QAGNN is its KG, which is based on the Disease Database portion of the Unified Medical Language System (UMLS) and DrugBank. The graph contains about 10k nodes and 44k edges, where the embeddings for each node are initialized using the biomedically

	Names				Gender+Ethnicity+Names									
	W	**A-A/B**	**H**	**As**	**M+W**	**M+A-A**	**M+B**	**M+H**	**M+As**	**F+W**	**F+A-A**	**F+B**	**F+H**	**F+As**
QAGNN	**10.5**	**10.5**	**12.6**	**10.5**	**9.3**	**14.2**	**11.5**	**9.8**	8.5	**9.3**	**15.0**	**11.5**	**12.5**	7.9
BioLinkBERT	7.4	6.0	8.5	6.0	8.8	11.9	9.8	8.1	**9.6**	8.5	11.5	10.3	10.0	**9.0**

Table 3.5: Percentage of questions with changed answers as compared to a question with no demographic information about the patient. M=male; F=female; W=White; B=Black; A-A=African-American; H=Hispanic; As=Asian; SOr=sexual orientation.

trained language model SapBERT [155]. SapBERT was trained using the UMLS vocabulary set 2020AA version, which contains biomedical synonyms from more than 150 controlled vocabularies, such as Gene Ontology and MeSH. QAGNN has 360M parameters.

For each answer choice of a given question, QAGNN first retrieves a subgraph from its KG using entity linking. That is, it finds entity mentions in the question and retrieves any entity in the main KG that appears in any 2-hop paths between pairs of found entities. Then, it concatenates the answer choice and question, followed by encoding using a LM. Next, it connects the encoded representation to the graph as a node. It then performs relevance scoring on each node in the created subgraph by concatenating it to the encoded representation node and calculating the likelihood using a LM. Lastly, using an attention-based graph neural network (GNN) module, it reasons over the graph to get a score for the answer choice. During the training procedure, it optimizes both the LM and its GNN end-to-end using cross-entropy loss. On the MedQA-USMLE dataset, SapBERT-based QAGNN achieves 38% accuracy.

3.1.4.2 BioLinkBert

The defining features of BioLinkBert are its pretraining method that incorporates document links and its LM which has similar hyperparameters to PubmedBERT [89] and is trained from scratch on the PubMed abstracts PubmedBERT is trained on. BioLinkBert has 340M parameters.

Given a corpus of text, BioLinkBert views it as a graph: it uses Pubmed Parser to extract citation links between documents and views the hyperlinks as edges. Then, to use the links in its LM

Model	Names									
	W		**B**		**A-A**		**H**		**AS**	
QAGNN	**38.6**		**39.5**		**39.5**		**39.3**		**38.5**	
BioLinkBert	38.2		37.3		37.3		37.6		37.6	
	M	**F**	**M**	**F**	**M**	**F**	**M**	**F**	**M**	**F**
QAGNN	38.6	38.1	**39.5**	**39.4**	**39.7**	**39.1**	**39.0**	**39.2**	**39.2**	**37.9**
BioLinkBert	**39.4**	**38.5**	38.6	37.0	35.1	38.7	36.8	37.2	37.3	35.4

Table 3.6: Accuracy when including names (rows 1 and 2) or names together with gender and ethnicity information (rows 3 and 4) for each model. *W*=White; *B*=Black; *A-A*=African-American; *H*=Hispanic; *As*=Asian;

	Random	**Gender**		**Ethnicity**					**SOr**			**Gender+Ethnicity**										**Gender+SOr**					
		M	**F**	**W**	**A-A**	**B**	**H**	**As**	**Hetero**	**Bi**	**Homo**	**M+W**	**M+A-A**	**M+B**	**M+H**	**M+As**	**F+W**	**F+A-A**	**F+B**	**F+H**	**F+As**	**M+Hetero**	**M+Bi**	**M+Homo**	**F+Hetero**	**F+Bi**	**F+Homo**
Generic	2	**17**	**16**	6	**14**	7	**9**	7	**9**	**11**	11	**11**	**11**	**12**	**13**	**8**	**13**	**13**	**13**	**12**	**12**	8	**10**	6	**12**	**15**	11
Biomedical	2	6	6	6	8	7	7	**11**	6	6	**14**	6	7	5	7	6	8	8	9	8	8	**9**	9	**23**	8	13	**23**

Table 3.7: Percentage of questions with changed answers between the biomedical and generic model as compared to a question with no demographic information about the patient. *M*=male; *F*=female; *W*=White; *B*=Black; *A-A*=African-American; *H*=Hispanic; *As*=Asian; *SOr*=sexual orientation.

	*O**	O	D	Gender		Ethnicity					SOr			Gender+Ethnicity										Gender+SOr					
				M	F	W	A-A	B	H	As	Hetero	Bi	Homo	M+W	M+A-A	M+B	M+H	M+As	F+W	F+A-A	F+B	F+H	F+As	M+Hetero	M+Bi	M+Homo	F+Hetero	F+Bi	F+Homo
1	*28.9*	26	25	29	27	27	26	26	27	27	28	27	26	27	27	27	26	25	27	27	27	26	24	27	27	28	25	26	25
2	***40***	**39**	**40**	**40**	**40**	**40**	**38**	**39**	**39**	**36**	**38**	**38**	**37**	**39**	**37**	**41**	**40**	**40**	**40**	**36**	**40**	**41**	**40**	**41**	**41**	**36**	**41**	**40**	**36**

Table 3.8: Accuracy (in percentages) of the biomedical and generic models on our demographically enhanced datasets. *M*=male; *F*=female; *W*=White; *B*=Black; *A-A*=African-American; *H*=Hispanic; *As*=Asian; *SOr*=sexual orientation; O*=original test dataset; O=the original, unmodified 100 questions; D=No demographic information; 1=Generic; 2=Biomedical.

pretraining procedure it places two documents which share a link in the same context, in addition to placing two random documents in the same context or a single document (contiguous). Next, it uses two self-supervised objectives. The first, masked language modeling, is common in many of the large LMs such as BERT [57]. In the second, document relation prediction, it classifies the link between the two documents as random, linked, or contiguous. On the MedQA-USMLE dataset, the base version of BioLinkBert achieves 40% accuracy while the large version achieves 44.6%. Here, we work with the base version because of its lower compute requirements.

3.1.5 Results

We look at two different effects of providing the model with irrelevant demographic information: 1) the percentage of questions for each model that change and 2) the accuracy change for each model. Note that these are not necessarily correlated: for example, accuracy does not change when initially incorrect answers change to other incorrect answers, or if the same numbers of answers change from incorrect to correct. It is also worth mentioning that **any change in model's answers is problematic, as these questions were verified to be independent of demographics.**

3.1.5.1 Changed Answers

Table 3.2 shows the percentage of questions for each model that change between each dimension's attribute and the dimensionless variation (e.g., between male and genderless).

The first column of Table 3.2, "Random", shows the result of our random change (Sec. 3.1.3). While the other values in the table are larger, and while the words "patient" and "person" may have different connotations for each model based on its training data, this suggests that, to some extent, random noise plays a role in the amount of change each model exhibits. Notably for gender, ethnicity, and sexual orientation, both models change around the same number of answers, except that BioLinkBert has a much higher number for Asian. Additionally, both models have almost double the amount of changed answers for homosexual than bisexual or heterosexual. For gender+ethnicity, QAGNN has an equivalent amount or more than BioLinkBert, though for gender+sexual orientation, BioLinkBert has more than double the amount for homosexuals, with a massive percentage of 23. We also examine the amount of answers for each model for gender, ethnicity, and sexual orientation, that change from being correct to incorrect, from incorrect to correct, and from incorrect to incorrect (Table 3.3). We can see that a model can have an increase in performance (see QAGNN males column which results in a 2% increase) while having the same number of answers change as a demographics which result in a decrease in performance (see QAGNN White column which result in a 4% decrease). This implies that accuracy alone is not sufficient to understand the effect irrelevant demographic information has on models' answer, and that further examination of the answers can contribute. For example, we see that adding most ethnicities results in 0 answers changing from incorrect to correct for QAGNN.

3.1.5.2 Changed Accuracy

While the reported accuracy on the original test dataset is 38% for QAGNN and 40% for BioLinkBert, the accuracy on our 100 randomly selected demographic-independent questions use to construct the vignettes is 40% for QAGNN and 39% for BioLinkBert. Table 3.4 shows our accuracy

results for each dimension for each algorithm.

As noted, accuracy change does not always correlate with answer change. For example from Table 3.2, while both models have about the same number of changed answers for gender, only QAGNN's accuracy for males is affected (increased by 2%). For ethnicity, both models' accuracy drops, with BioLinkBert's accuracy by 3% for Asian and QAGNN's accuracy by 4% for Black. Sexual orientation improves BioLinkBert performance on bisexual and decreases QAGNN's on every variation. Gender+ethnicity decreases QAGNN performance the most (up to 6%), while gender+sexual orientation improves BioLinkBert's performance on any variation except for homosexual.

3.1.6 Analysis: Names

Similarly to the above experiments, we also evaluate the effect names have on the two types of models. For names by themselves, for each ethnicity (Black, White, Hispanic, Asian) we use the corresponding 20 names (10 for males and 10 for females). For names+ethnicity+gender, we split the names into their ethnicity and gender.

Table 3.5 and 3.6 show our results: Tables 3.5 displays the number of changed answers, while Table 3.6 shows accuracy changes. We can see that names alone have a moderate effect on the performance of both models, decreasing the performance in any variation by up to 1.65%. From our baseline experiment this may be due to random noise. However, by looking at the number of changed answers, we can see that both models have the most change for Hispanics, with QAGNN change of up to 12.6% and BioLinkBert by up to 8.5%. Interestingly, QAGNN has the same number of changed answers for White, Black, and Asian, but a different number for Hispanic. More results can be seen in the combination of gender, ethnicity, and names, in which the performance can decrease by up to 3.9% for BioLinkBert in African American males, and by up to 2.1% for QAGNN in Asian females. However, the amount of changed answers is up to 15% in QAGNN for African American females and up to 11.9% for BioLinkBert in African American males. This implies that even though both models were trained on PubMed data, irrelevant information like names affect them, which is highly problematic.

3.1.7 Medical vs. Generic LMs

In addition to our main results, we also compare how the performance of a biomedically-trained transformer differs from that of a generic one. In particular, we use the same code for the BioLinkBert QA system, but instead of using the medically-trained base (trained from scratch on PubMed abstracts), we use a transformer which is trained on generic English text.

Similar to our analysis between QAGNN and BioLinkBert above, our analysis between the biomedical and generic models can be split into the amount of answers and accuracy that changes when the dimensions change. From Table 3.7 it is visible that the generic transformer has more than double the amount of answers change for each gender. It also has an equivalent amount or more for almost any ethnicity, except for Asians. Notably, for sexual orientation, the generic transformer has almost double the amount of answers change for bisexuals, while the biomedical transformer has more for homosexuals. The generic transformer has significantly larger values than the biomedical transformer in any gender+ethnicity combination, while for gender+sexual orientation, the biomedical system has significantly larger values for homosexuals. From Table 3.8 it is clear that BioLinkBert significantly outperforms its generic LM variation. From the change in accuracy we can see that, while the biomedical transformer's accuracy increases when gender is removed ("no info"), the generic transformer's accuracy decreases. We can also see that the biomedical transformer's accuracy changes more for ethnicity and sexual orientation, while the generic model changes more for gender.

3.2 Comparing Template-based and Template-free Language Model Probing

Two common approaches for probing models' parametric knowledge are template-based prompts, where manually crafted patterns (e.g., "X was born in 1994") are used to query the model, and template-free prompts, which rely on naturally occurring text from contexts. **The work described in this section has been published in EACL 2024 [256]**.

Template-based Probing	Template-free Probing
Template: "[X] (born [MASK])"	N/A
Peter F. Martin (born [MASK])	Peter F Martin (born [MASK]) is an American politician [...]
Dennis B. Sullivan (born [MASK])	Sullivan was born in Chippewa Falls Wisconsin in [MASK]
Tan Jiexi (born [MASK])	Tan Jiexi (born December 2, [MASK] in Shenzhen,China), is a Chinese singer-songwriter
Tasos Neroutsos (born [MASK])	Neroutsos was born in Athens in [MASK] to a wealthy family

Table 3.9: Template-based and template-free probing examples from the LAMA: Google-RE dataset. In the template-based approach, each template is used to create many prompts which are identical except for the subject entity. In comparison, the template-free approach does not use templates and prompts LMs with an often unique prompt per entity.

3.2.1 Introduction

In the past few years there has been a growing interest in understanding what parametric knowledge LMs contain [113]. One standard approach of probing LMs for knowledge consists of using "fill-in-the-blank" cloze statements [269, 126, 281, 219, 180], where models are tasked with predicting a masked entity given a prompt, e.g., "Dante was born in [MASK]".

While this has been studied in various settings, including multilingual [125], single token predictions [345, 219, 281, 26], multi-token predictions [125], and prompt optimization [345, 269], the differences between the two types of prompts – template-based and template-free; see Table 3.9 – have been overlooked so far.

Although in both methods whether a LM knows a fact is defined by its ability to successfully predict the masked object in a prompt [219], the template-based approach uses templates (which are often manually created) to create the prompts, where *each template is used to create many prompts which are identical except for the subject entity.* In comparison, the template-free approach does not use templates and prompts LMs with an often *unique prompt per entity.*

Each of the two methods has its pros and cons. For example, while the template-based approach generally guarantees that the prompt is evaluating the required knowledge, it requires expensive domain experts. And while the prompts in the template-free approach are more similar to the training data LMs are trained on, as they come from real-world text and not from artificial

templates, they may contain additional irrelevant information.

We hypothesize that this may result in different rankings for the same models when being prompted via the two approaches.

Here, we 1) evaluate 16 different LMs on 10 probing datasets (4 template-based and 6 template-free) in multiple domains; 2) we propose a method to create template-free domain-specific datasets and use it to develop the first template-free biomedical probing dataset, which allows us to compare the effect of the two probing approaches in two different domains; 3) ask the following research questions: (RQ1) Do model rankings differ between template-based and template-free probing? (RQ2) Do models' absolute scores differ between the two approaches? (RQ3) Do the answers to the two previous questions differ between general and domain-specific models?

Our study's results can be summarized as follows: 1) There is a discrepancy in ranking models between template-free and template-based methods, except for the top domain-specific models. 2) Scores decrease by up to 42% Acc@1 when comparing parallel template-free and template-based prompts (i.e., similar subject entities and masked objects). 3) Perplexity is negatively correlated with accuracy in the template-free approach, but, counter-intuitively, they are positively correlated for template-based probing. 4) Models have a tendency to predict similar objects to various prompts, even when the subjects change, when utilizing template-based probing, which is less common when employing template-free techniques.

3.2.2 Experiments

The LMs' input for both template-free and template-based probing is a prompt with one masked entity; see Table 5.2. The models are then tasked with predicting the masked entity.

While most previous work focuses solely on single-token mask prediction, many entities are composed of more than one token. Hence, we follow [125] that expand the probing technique to multi-token prediction and show that it is a better method to investigate knowledge captured by LMs. In particular, we use the same alternative to the "fill-in-the-blank" querying by framing the task as entity ranking. However, while [125] limit the prediction to entities of the type required by

the prompt, we relax this limitation because 1) few existing datasets have entity type information, and using external resource to classify entity types may be inaccurate and skew results; 2) this simplifies the problem for the models which, again, may skew results. Instead, we allow our models to predict any entity from the dataset's entity list.

Lastly, we experiment in both general and biomedical domains to analyze whether our experiments generalize. Hence, a key portion of the experiments pertains to biomedical models, as in addition to generic English models, we use both biomedical fine-tuned models and biomedical models.

Evaluation Metric Following prior work by [281], we use top-k accuracy (Acc@k), wherein a score of 1 is given if the correct entity appears among the top k predicted entities, and 0 otherwise. Since entities are often related to numerous other entities (N-to-M connections), we use Acc@1, Acc@5, and Acc@10.

3.2.2.1 Models

We evaluate 16 different LMs belonging to 3 categories: 1) trained exclusively on generic English text; 2) pretrained on generic English text and fine-tuned on biomedical text; 3) trained only on biomedical text; see Table 3.10 for an overview.

Generic English Models We experiment with 7 generic English models: DistilBERT [247], BERT-base/large [57], RoBERTa-base/large [159], ALBERT-base/large [140].

Biomedical Fine-tuned Models We probe 6 models which have been pretrained on generic text, followed by finetuning: PMC RoBERTa[2] , COVID Bert,[3] BlueBert [217], Bio Discharge Summary BERT [7], Bio ClinicalBERT [7], and BioMed-RoBERTa [93].

Biomedical Models We further experiment with PubMedBERT [89], Bioformer [71], and BioM-ELECTRA [6].

[2] https://huggingface.co/raynardj/pmc-med-bio-mlm-roberta-large
[3] https://huggingface.co/mrm8488/bioclinicalBERT-fine-tuned-covid-papers

Model	Parameters	Data
⋆PubMedBERT	109M	PubMed abstracts+PMC full-text articles (3.2B words/21GB)
⋆Bioformer	42M	33M PubMed abstracts+1M PMC full-text articles
⋆BioM-ELECTRA-Generator	49M	PubMed Abstracts
†BioMed-RoBERTa	124M	RoBERTa (160GB)+Semantic Scholar corpus (2.68M papers/47GB)
†COVID Bert	108M	N/A
†BlueBert	109M	Bert+PubMed abstracts+MIMIC-III clinical notes (4500M words/27GB)
†Bio Discharge Summary BERT	108M	Biobert (18B words)+MIMIC III discharge summaries (880M words)
†PMC RoBERTa	355M	RoBERTa (160GB)+ PMC and PubMd abstracts
†Bio ClinicalBERT	108M	Biobert (18B words)+MIMIC notes (880M words)
◇RoBERTa-base	124M	BookCorpus, English Wikipedia, CC-News, OpenWebText, Stories (160GB)
◇RoBERTa-large	355M	BookCorpus, English Wikipedia, CC-News, OpenWebText, Stories (160GB)
◇BERT-base	109M	BookCorpus, English Wikipedia (16GB)
◇BERT-large	334M	BookCorpus, English Wikipedia (16GB)
◇ALBERT-base	12M	BookCorpus, English Wikipedia (16GB)
◇ALBERT-base	18M	BookCorpus, English Wikipedia (16GB)
◇DistilBERT	12M	BookCorpus, English Wikipedia (16GB)

Table 3.10: Models, number of parameters, and their training data. # Parameters were taken directly from the Huggingface implementation. GB/# words are taken from the authors' reports; *N/A*=no information regarding the training data has been provided by the authors. In blue (⋆) we have models that were only trained on biomedical text. In green (†) we have models that were trained on generic English text and fine-tuned on biomedical text. In red (◇) we have models that were only trained on generic English text.

3.2.2.2 Template-based Probing

Comparative Toxicogenomics Database The Comparative Toxicogenomics Database (CTD) is a biomedical database with relations and interactions between biomedical entities. We use the same subset as [281], which contains template-based prompts that were manually curated.

Biomedical Wikidata The Wikidata dataset from [281] contains template-based prompts and is based on a general knowledge base. We use the same subset of it as [281] which only contains biomedical entities and relations that were manually curated.

Google-RE (Templates) Google-RE [219] contains 6.11K template-based prompts from Wikipedia and 3 relations.

T-REx (Templates) The T-REx dataset from [219] is based on a subset of Wikidata triples and contain 41 relations. The authors manually define a template for each relation which result in 1.3M template-based prompts.

3.2.2.3 Template-free Probing

Google-RE (Template-free) While the Google-RE dataset from [219] contains 6.11K template-prompts from Wikipedia, **each prompt is manually aligned by the creators of the dataset to text** from Wikipedia that supports it. We use the latter as template-free prompts; see Table 5.2 for examples.

T-REx (Template-free) While the T-REx dataset from [219] contains 1.3M templates from Wikidata, **each prompt is automatically aligned by the creators of the dataset to natural text** from Wikipedia that supports it and which we use for template-free probing.

ConceptNet The ConceptNet dataset [219] contains 29.8K natural prompts from Open Mind Common Sense, covering 16 relations.

SQuAD The SQuAD dataset from [219] contains 305 template-free prompts from the SQuAD dataset [233]: the authors select a subset of 305 context-insensitive questions from the SQuAD validation set and manually modify them to be a cloze-style question.

LIPID We further experiment with our novel tempLate-free bIomedical ProbIng Dataset (LIPID), composed of 88,666 template-free prompts from PubMed abstracts which we split into two datasets: chemicals and genes. The chemical portion contains 46,827 chemical-related prompts and 1870 unique chemical entities, where the gene portion contains 41,839 gene-related prompts and 2591 unique gene entities. While entity-centric cloze-style QA datasets have previously been proposed for biomedicine, such as BioRead [213] and BioMRC [214], *we create the first template-free dataset for biomedical probing, which allows us to compare the effect of the two probing approaches in 2 different domains.* Furthermore, to encourage more research on template-free probing, we propose an approach to develop such domain-specific datasets composed of four steps.

3.2.2.4 The Creation of LIPID

We create LIPID, a template-free dataset for probing models with prompts from the biomedical domain, via four steps, which we describe below: 1) retrieving a collection of biomedical text, 2) using a list of biomedical entities to select sentences, 3) filtering the resulting sentences using a list of keywords, and 4) entity masking. The creation of our dataset *takes about a day*, which consists of automatically downloading six months worth of PubMed publications, parsing, filtering, and masking. **Biomedical Text Retrieval** It is important to ensure that the LMs were not trained on the test data used for probing. For that, we choose to use PubMed abstracts[4] which were submitted after December 2021, which is the publication date of the most recent Biomedical LM we will probe. Such separation between the dates ensures that our questions and contexts which are used to prompt the LMs for knowledge are **entirely unseen** to all our models during training.

Biomedical Entities We use a list of 1870 unique chemicals and 2591 unique genes taken from the ChemDNER dataset [135] and retrieve sentences that include exactly one of those entities.

Quality Control Since we care about sentences that are facts (e.g., "Penicillin is used to treat certain infections") rather than hypotheses, suppositions, or various other sentence forms (e.g., "We examine the effect penicillin has on infections"), we filter the resulting sentences from the

[4] https://ftp.ncbi.nlm.nih.gov/pubmed/updatefiles/

previous step using a simple list of keywords we create. For example, we remove all sentences that contain parentheses, as often the entity will precede its short notation (e.g., "penicillin (PCN)") which will most likely reveal to the model the identity of the masked entity. The simple list of keywords is: "here", "we ", "investigate", "study", "propose", "outline", "(", "our ", "performed", "suggest", and "However." Finally, two annotators – one of which is a medical expert and the second is a CS PhD student – review 200 random prompts and evaluate the number of non-factual statements. For the chemical portion of the data the average is 92.5, and for the gene portion the average is 96.0.

Masking Lastly, we mask the entity in each sentence with a masking token. For example, the sentence "Penicillin is used to treat infections" becomes "[MASK] is used to treat infections."

LIPID Statistics Example prompts from our datasets are as follows. From the chemical portion: "A key to longevity assurance is the nutrient-sensing [MASK] pathway". From the gene portion: "Amyloid-β is a product of the processing of the amyloid precursor protein, encoded by the [MASK] gene on chromosome 21". On average, each chemical and gene entity appears 25.04 and 16.14 times, respectively, with standard deviations of 91.85 and 57.58. The maximum number of times any chemical or gene entity appears is 1669 and 1541 times, respectively. The minimum number of times any entity appears is 1 for both chemicals and genes.

3.2.3 Results

Tables 3.11 and 3.12 show our main results on the template-based and template-free datasets.

Model Rankings In both the template-based and template-free biomedical datasets – CTD, Biomed-Wikidata, and our novel LIPID datasets – PubMedBERT performs best, followed by Bioformer and BioM-ELECTRA: both techniques clearly separate models that were trained solely on biomedical data (in blue) as opposed to those who were fine-tuned on biomedicine (in green) or general domain (in red). In comparison, on general-domain datasets, BERT-large performs best, followed by either BERT-base or DistilBERT. Furthermore, the top-5 general-domain models are roughly the same across all general-domain datasets. However, as the rank increases, the pattern is less obvious and there is no clear separation between models in blue and those in green, e.g.,

while PubMedBERT is the 6th best, followed by Bioformer as 7th or 8th, its rank changes to 15 (i.e., second to last) on Google-RE template-based. This is especially surprising as on the same Google-RE dataset, but in the template-free setting, Bioformer ranks 7th. Similarly, large changes in ranking can be seen for RoBERTa-base, which moves from rank 12 for template-free to 7 in the template-based Google-RE. This is also visible for the T-REx dataset, where RoBERTa-large moves from rank 7 in the template-free to 12 in the template-based setting. Similar ranking differences between datasets also appear in general models that are fine-tuned on biomedical data (in green). Notably, model rankings change between two general-domain datasets (e.g., Google-RE and T-REx), between domain-specific datasets (e.g., CTD and Biomed-Wikidata), and between both the template-free and template-based approaches (e.g., both Google-RE and T-REx settings).

Model Scores Since both Google-RE and T-REx are composed of parallel template-free and template-based datasets in which each template has a corresponding template-free text, see Table 5.2, we can directly compare models' scores across them.

We find substantial different scores between the template-free and template-based datasets. For example, the average Acc@1 on the template-free datasets Google-RE and T-REx are 0.094 and 0.21, respectively. These scores change to 0.025 and 0.11 when the dataset is converted to template-based.

We see the largest performance difference in BERT-large, which obtains a score of 0.72 Acc@1 on template-free T-REx, but a score of 0.3 Acc@1 on the corresponding template-based data.

While T-REx and Google-RE are the only parallel datasets we have, allowing us to directly compare between the datasets, we can also see that the scores are different in general between template-based and template-free datasets: e.g., the average Acc@1 on the CTD and Biomed-Wikidata are 0.002 and 0.011, while on our LIPID datasets of the same biomedical domain, the average Acc@1 are 0.12 on genes and 0.18 on chemicals.

Another strange model behavior we see for the template-based datasets is the effect of model size: larger models generally perform better than their smaller counterparts. This can be seen across all base and large models on all template-free datasets. However, this is not the case in the

Model	Google-RE				T-REx				Biomed-Wikidata				CTD			
	A@1	A@5	A@10	R	A@1	A@5	A@10	R	A@1	A@5	A@10	R	A@1	A@5	A@10	R
⋆PubMedBERT	$5.4e^{-3}$	$1.7e^{-2}$	$3.2e^{-2}$	8	$1.1e^{-1}$	$2.1e^{-1}$	$2.7e^{-1}$	6	$4.4e^{-2}$	$1.3e^{-1}$	$1.9e^{-1}$	1	$7.8e^{-3}$	$2.9e^{-2}$	$4.5e^{-2}$	1
⋆Bioformer	$9.8e^{-4}$	$6.2e^{-3}$	$1.3e^{-2}$	15	$9.3e^{-2}$	$1.7e^{-1}$	$2.1e^{-1}$	7	$3.7e^{-2}$	$1.0e^{-1}$	$1.6e^{-1}$	2	$6.0e^{-3}$	$2.3e^{-2}$	$3.6e^{-2}$	2
⋆BioM-ELECTRA	$1.3e^{-3}$	$7.8e^{-2}$	$1.5e^{-2}$	13	$5.4e^{-2}$	$1.3e^{-1}$	$1.8e^{-1}$	9	$3.0e^{-2}$	$1.0e^{-1}$	$1.5e^{-1}$	3	$3.4e^{-3}$	$1.4e^{-2}$	$2.6e^{-2}$	3
†BioMed-RoBERTa	$1.2e^{-2}$	$3.0e^{-2}$	$4.3e^{-2}$	6	$5.1e^{-2}$	$1.2e^{-1}$	$1.7e^{-1}$	11	$2.3e^{-3}$	$1.4e^{-2}$	$3.4e^{-2}$	12	$3.6e^{-4}$	$3.7e^{-3}$	$8.4e^{-3}$	6
†COVID Bert	$9.8e^{-4}$	$4.5e^{-3}$	$1.2e^{-2}$	16	$2.8e^{-2}$	$6.8e^{-2}$	$1.1e^{-1}$	13	$8.1e^{-3}$	$4.3e^{-2}$	$6.4e^{-2}$	6	$4.0e^{-3}$	$1.0e^{-2}$	$2.0e^{-2}$	4
†BlueBert	$1.6e^{-3}$	$9.9e^{-3}$	$1.6e^{-2}$	12	$3.5e^{-2}$	$1.0e^{-1}$	$1.5e^{-1}$	14	$1.3e^{-2}$	$5.1e^{-2}$	$8.7e^{-2}$	4	$5.4e^{-4}$	$5.4e^{-3}$	$9.8e^{-3}$	12
†Discharge BERT	$9.8e^{-4}$	$7.2e^{-3}$	$1.3e^{-2}$	14	$2.1e^{-2}$	$6.2e^{-2}$	$9.7e^{-2}$	15	$6.8e^{-3}$	$3.7e^{-2}$	$6.0e^{-2}$	10	$3.9e^{-3}$	$1.2e^{-2}$	$2.0e^{-2}$	5
†PMC RoBERTa	$3.2e^{-3}$	$1.7e^{-2}$	$3.5e^{-2}$	10	$4.0e^{-2}$	$9.4e^{-2}$	$1.3e^{-1}$	12	$2.0e^{-3}$	$1.7e^{-2}$	$3.0e^{-2}$	14	$8.1e^{-4}$	$3.7e^{-3}$	$8.5e^{-3}$	11
†Bio ClinicalBERT	$1.9e^{-3}$	$5.0e^{-3}$	$1.0e^{-2}$	11	$1.2e^{-2}$	$4.1e^{-2}$	$6.6e^{-2}$	16	$7.8e^{-3}$	$3.6e^{-2}$	$6.2e^{-2}$	7	$1.9e^{-3}$	$1.0e^{-2}$	$1.5e^{-2}$	7
◇RoBERTa-base	$6.3e^{-3}$	$2.5e^{-2}$	$4.9e^{-2}$	7	$5.3e^{-2}$	$9.4e^{-2}$	$1.2e^{-1}$	9	$1.3e^{-3}$	$2.2e^{-2}$	$3.6e^{-2}$	15	$3.6e^{-4}$	$4.4e^{-3}$	$7.9e^{-3}$	16
◇RoBERTa-large	$4.2e^{-3}$	$2.3e^{-2}$	$4.2e^{-2}$	9	$5.6e^{-2}$	$1.1e^{-1}$	$1.5e^{-1}$	8	$2.0e^{-3}$	$2.0e^{-2}$	$3.7e^{-2}$	13	$5.4e^{-4}$	$3.7e^{-3}$	$7.4e^{-3}$	13
◇BERT-base	$1.1e^{-1}$	$2.3e^{-1}$	$3.2e^{-1}$	2	$3.0e^{-1}$	$5.3e^{-1}$	$6.4e^{-1}$	2	$7.8e^{-3}$	$3.4e^{-2}$	$6.0e^{-2}$	8	$8.1e^{-4}$	$3.9e^{-3}$	$9.6e^{-3}$	10
◇BERT-large	$1.1e^{-1}$	$2.4e^{-1}$	$3.3e^{-1}$	1	$3.0e^{-1}$	$5.3e^{-1}$	$6.5e^{-1}$	1	$7.5e^{-3}$	$3.9e^{-2}$	$5.2e^{-2}$	9	$1.3e^{-3}$	$5.3e^{-3}$	$1.1e^{-2}$	8
◇ALBERT-base	$2.2e^{-2}$	$7.4e^{-2}$	$1.2e^{-1}$	5	$1.4e^{-1}$	$3.0e^{-1}$	$4.2e^{-1}$	5	$3.0e^{-3}$	$3.3e^{-2}$	$5.6e^{-2}$	11	$4.5e^{-4}$	$3.8e^{-3}$	$1.0e^{-2}$	14
◇ALBERT-large	$2.6e^{-2}$	$8.4e^{-2}$	$1.2e^{-1}$	4	$2.1e^{-1}$	$4.0e^{-1}$	$5.2e^{-1}$	4	$1.0e^{-3}$	$1.9e^{-2}$	$3.5e^{-2}$	16	$4.5e^{-4}$	$2.2e^{-3}$	$6.6e^{-3}$	15
◇DistilBERT	$1.0e^{-1}$	$2.1e^{-1}$	$2.9e^{-1}$	3	$2.7e^{-1}$	$5.2e^{-1}$	$6.4e^{-1}$	3	$1.1e^{-2}$	$4.2e^{-2}$	$6.8e^{-2}$	5	$9.0e^{-4}$	$4.7e^{-3}$	$1.1e^{-2}$	9
Average	$2.5e^{-2}$	$6.6e^{-2}$	$9.1e^{-2}$		$1.1e^{-1}$	$2.1e^{-1}$	$2.8e^{-1}$		$1.1e^{-2}$	$4.6e^{-2}$	$7.3e^{-2}$		$2.0e^{-3}$	$8.6e^{-3}$	$1.5e^{-2}$	

Table 3.11: **Template-based** results. We report Acc@1/Acc@5/Acc@10 of each model and the macro average, the ranking of it based on its Acc@1 score (or Acc@5/10 if there is a tie) for each dataset column "(R)". A=Acc.

template-based datasets: e.g., on CTD and Biomed-Wikidata, ALBERT-base outperforms its larger counterpart, and, on Google-RE, RoBERTa-base outperforms RoBERTa-large. We can further see this pattern with BERT on the Biomed-Wikidata dataset, where on T-REx and Google-RE the BERT-base version performs as well as BERT-large.

3.2.4 Discussion and Analysis

We now discuss and investigate why the aforementioned differences in scores and rankings occur between datasets and the two probing approaches.

3.2.4.1 Vocabulary

One obvious reason for the difference in scores between the datasets is the vocabulary of the models. For example, models trained on PubMed full text articles (e.g., PubMedBert and Bioformer) might have very uncommon chemical and gene names in their vocabularies in comparison to models trained on generic English text. This may result in biomedical models scoring lower on general domain datasets, and vice versa. However, framing both probing methods as the entity ranking method described in Section 3.2.2 by averaging log probabilities of the individual tokens of each

Model	Google-RE				ConceptNet				SQuAD				T-REx				LIPID-Gene				LIPID-Chem			
	A@1	A@5	A@10	R	A@1	A@5	A@10	R	A@1	A@5	A@10	R	A@1	A@5	A@10	R	A@1	A@5	A@10	R	A@1	A@5	A@10	R
⋆PubMedBERT	$1.0e^{-1}$	$2.3e^{-1}$	$2.9e^{-1}$	68	$8.9e^{-2}$	$1.7e^{-1}$	$2.1e^{-1}$	61	$8.8e^{-1}$	$3.7e^{-1}$	$4.5e^{-1}$	61	$9.9e^{-1}$	$3.4e^{-1}$	$4.0e^{-1}$	64	$3.3e^{-1}$	$5.7e^{-1}$	$6.1e^{-1}$	14	$8.8e^{-1}$	$6.4e^{-1}$	$6.9e^{-1}$	1
⋆Bioformer	$6.3e^{-2}$	$1.5e^{-1}$	$1.9e^{-1}$	75	$2.2e^{-2}$	$1.2e^{-1}$	$1.5e^{-1}$	81	$2.2e^{-2}$	$7.7e^{-2}$	$3.4e^{-1}$	81	$4.4e^{-2}$	$2.7e^{-2}$	$3.1e^{-1}$	83	$8.8e^{-2}$	$5.2e^{-1}$	$5.7e^{-1}$	24	$3.3e^{-1}$	$6.1e^{-1}$	$6.6e^{-1}$	2
⋆BioM-ELECTRA	$5.1e^{-2}$	$1.0e^{-1}$	$1.4e^{-1}$	85	$1.1e^{-2}$	$1.1e^{-1}$	$1.4e^{-1}$	91	$0.0e^{-1}$	$2.6e^{-1}$	$3.2e^{-1}$	11	$2.2e^{-2}$	$2.3e^{-2}$	$2.8e^{-1}$	13	$6.6e^{-1}$	$4.8e^{-1}$	$5.2e^{-1}$	34	$3.3e^{-1}$	$6.0e^{-1}$	$6.5e^{-1}$	3
†BioMed-RoBERTa	$4.6e^{-2}$	$1.1e^{-1}$	$1.6e^{-1}$	13	$4.6e^{-2}$	$6.4e^{-2}$	$8.2e^{-2}$	11	$2.2e^{-2}$	$6.6e^{-1}$	$3.3e^{-1}$	9	$1.2e^{-2}$	$4.4e^{-1}$	$3.0e^{-1}$	14	$1.1e^{-2}$	$8.8e^{-2}$	$1.1e^{-1}$	12	$7.2e^{-2}$	$5.8e^{-2}$	$7.8e^{-2}$	13
†COVID Bert	$3.2e^{-3}$	$1.7e^{-2}$	$3.4e^{-2}$	15	$9.2e^{-2}$	$1.2e^{-1}$	$1.7e^{-1}$	78	$2.2e^{-2}$	$2.2e^{-2}$	$2.7e^{-1}$	16	$6.6e^{-2}$	$1.5e^{-1}$	$1.9e^{-1}$	11	$3.3e^{-1}$	$2.0e^{-1}$	$2.3e^{-1}$	42	$2.2e^{-1}$	$3.2e^{-1}$	$3.6e^{-1}$	4
†BlueBert	$3.2e^{-3}$	$1.6e^{-2}$	$3.4e^{-2}$	13	$5.5e^{-2}$	$8.2e^{-2}$	$1.1e^{-1}$	14	$9.9e^{-2}$	$1.5e^{-1}$	$2.2e^{-1}$	13	$5.5e^{-2}$	$8.9e^{-2}$	$1.3e^{-1}$	15	$9.9e^{-2}$	$1.0e^{-1}$	$1.2e^{-1}$	11	$4.4e^{-1}$	$2.1e^{-1}$	$2.4e^{-1}$	10
†Discharge BERT	$9.4e^{-4}$	$7.7e^{-3}$	$1.6e^{-2}$	14	$9.9e^{-2}$	$1.1e^{-1}$	$1.5e^{-1}$	15	$9.9e^{-2}$	$1.6e^{-1}$	$2.2e^{-1}$	14	$3.3e^{-2}$	$9.7e^{-2}$	$1.2e^{-1}$	11	$0.0e^{-1}$	$1.5e^{-1}$	$1.7e^{-1}$	51	$8.8e^{-1}$	$2.6e^{-1}$	$3.0e^{-1}$	7
†PMC RoBERTa	$4.9e^{-2}$	$1.1e^{-1}$	$1.8e^{-1}$	9	$3.0e^{-2}$	$6.0e^{-2}$	$7.8e^{-2}$	$1.1e^{-1}$	$2.7e^{-1}$	$3.5e^{-1}$	$1.3e^{-1}$	$2.5e^{-1}$	$3.2e^{-1}$	$2.9e^{-2}$	$5.7e^{-2}$	$7.2e^{-2}$	$2.1e^{-2}$	$4.3e^{-2}$	$5.7e^{-2}$	15				
†Bio ClinicalBERT	$1.8e^{-3}$	$7.9e^{-3}$	$1.3e^{-2}$	13	$5.5e^{-2}$	$8.3e^{-2}$	$1.1e^{-1}$	13	$6.6e^{-2}$	$1.1e^{-1}$	$1.8e^{-1}$	13	$6.6e^{-2}$	$8.1e^{-2}$	$1.1e^{-1}$	16	$9.9e^{-2}$	$1.0e^{-1}$	$1.2e^{-1}$	11	$3.3e^{-1}$	$2.0e^{-1}$	$2.3e^{-1}$	11
◇RoBERTa-base	$3.2e^{-2}$	$9.1e^{-2}$	$1.4e^{-1}$	11	$9.9e^{-2}$	$4.4e^{-2}$	$5.8e^{-2}$	$1.1e^{-1}$	$0.0e^{-1}$	$2.0e^{-1}$	$3.0e^{-1}$	9	$0.0e^{-2}$	$1.9e^{-1}$	$2.5e^{-1}$	11	$8.8e^{-2}$	$3.9e^{-2}$	$5.0e^{-2}$	11	$2.2e^{-2}$	$2.7e^{-2}$	$3.6e^{-2}$	16
◇RoBERTa-large	$4.5e^{-2}$	$1.2e^{-1}$	$1.9e^{-1}$	13	$5.5e^{-2}$	$7.2e^{-2}$	$9.3e^{-2}$	1.1	$4.4e^{-3}$	$0.0e^{-1}$	$4.0e^{-1}$	7	$1.4e^{-2}$	$2.7e^{-1}$	$3.3e^{-1}$	7	$2.9e^{-2}$	$5.6e^{-2}$	$7.1e^{-2}$	12	$1.1e^{-2}$	$4.4e^{-2}$	$5.9e^{-2}$	14
◇BERT-base	$2.5e^{-1}$	$4.6e^{-1}$	$5.7e^{-1}$	21	$4.4e^{-1}$	$2.8e^{-1}$	$3.4e^{-1}$	23	$6.6e^{-1}$	$6.9e^{-1}$	$8.1e^{-1}$	3	$5.2e^{-1}$	$7.8e^{-1}$	$8.4e^{-1}$	26	$7.2e^{-2}$	$9.5e^{-2}$	$1.0e^{-1}$	11	$8.8e^{-2}$	$2.5e^{-1}$	$2.8e^{-1}$	8
◇BERT-large	$2.7e^{-1}$	$4.8e^{-1}$	$5.9e^{-1}$	11	$7.2e^{-1}$	$3.0e^{-1}$	$3.7e^{-1}$	14	$4.4e^{-1}$	$7.7e^{-1}$	$8.6e^{-1}$	5	$5.6e^{-1}$	$8.0e^{-1}$	$8.6e^{-1}$	17	$5.5e^{-2}$	$1.0e^{-1}$	$1.2e^{-1}$	92	$0.0e^{-1}$	$2.7e^{-1}$	$3.0e^{-1}$	5
◇ALBERT-base	$1.4e^{-3}$	$3.2e^{-1}$	$4.1e^{-1}$	51	$1.1e^{-2}$	$2.2e^{-2}$	$2.8e^{-1}$	52	$5.5e^{-1}$	$5.3e^{-1}$	$6.2e^{-1}$	5	$3.3e^{-1}$	$5.8e^{-1}$	$6.8e^{-1}$	5	$7.6e^{-2}$	$1.3e^{-1}$	$1.6e^{-1}$	81	$2.2e^{-1}$	$2.0e^{-1}$	$2.5e^{-1}$	12
◇ALBERT-large	$2.0e^{-1}$	$3.9e^{-1}$	$4.8e^{-1}$	41	$4.4e^{-1}$	$2.7e^{-1}$	$3.3e^{-1}$	33	$0.0e^{-1}$	$5.8e^{-1}$	$7.1e^{-1}$	4	$4.1e^{-1}$	$6.6e^{-1}$	$7.5e^{-1}$	4	$9.6e^{-2}$	$1.6e^{-1}$	$1.9e^{-1}$	61	$4.4e^{-1}$	$2.3e^{-1}$	$2.8e^{-1}$	9
◇DistilBERT	$2.5e^{-1}$	$4.6e^{-1}$	$5.6e^{-1}$	31	$3.2e^{-1}$	$2.7e^{-1}$	$3.4e^{-1}$	43	$8.8e^{-1}$	$7.3e^{-1}$	$8.2e^{-1}$	2	$5.0e^{-1}$	$7.6e^{-1}$	$8.3e^{-1}$	3	$8.8e^{-2}$	$1.3e^{-1}$	$1.5e^{-1}$	71	$9.9e^{-1}$	$2.8e^{-1}$	$3.2e^{-1}$	6
Average	$9.4e^{-2}$	$1.9e^{-1}$	$2.4e^{-1}$		$7.3e^{-2}$	$1.4e^{-1}$	$1.8e^{-1}$		$1.7e^{-1}$	$3.6e^{-1}$	$4.5e^{-1}$		$2.1e^{-1}$	$3.6e^{-1}$	$4.2e^{-1}$		$1.2e^{-1}$	$1.8e^{-1}$	$2.1e^{-1}$		$1.8e^{-1}$	$2.6e^{-1}$	$2.9e^{-1}$	

Table 3.12: **Template-free** results. We report Acc@1/Acc@5/Acc@10 of each model and the macro average, the ranking of it based on its Acc@1 score (or Acc@5/10 if there is a tie) for each dataset column "(R)". A=Acc.

entity should mitigate this effect. And while the vocabulary may have some effect on models' scores, it is important to note that on the two parallel datasets – Google-RE and T-REx, all models stay the same and the only different variable is the probing method. However, the ranking still changes.

3.2.4.2 Reused Templates

Another possible reason for the difference between the probing techniques is the variability of the templates. As experts are often required to create templates, the number of different templates is low. For example, the Google-RE dataset is composed of five templates, but [219] only use three of these. This may skew results, as, e.g., models may score lower on templates they are not familiar with.

To further analyze this, we evaluate the average Acc@10 over all models on each of the three templates [219] use from the Google-RE template-based dataset. We find that the score on the template "ENTITY (born [MASK])" is 8.9, the score on the template "ENTITY was born in [MASK]" 0.0, and the score on the template "ENTITY died in [MASK]" is 0.03. This highlights that models do in fact struggle with some templates more than with others.

3.2.4.3 Different Data Distribution

The data distribution of the templates differs from real-world text which LMs are often trained on. For example, the template-based prompts in Table 5.2 are not full sentences, which standard LMs are trained on. This by itself may result in skewed results, but, in general, models trained on the masked language modeling task may be getting the answer wrong because they are unfamiliar with such text structure, rather than lacking knowledge of the domain itself (See Section 3.2.4.2).

3.2.4.4 Overconfident Models

We note from Table 4 in [281]'s paper that models often make the same predictions for a given template, even when the entities change (e.g., on the CTD dataset ESR1 is always first, followed by NR1I2). Manually analyzing such behavior, we find similar patterns in our various datasets. Additionally, we find that models that do not do that (e.g., Bioformer) often perform much better than those models that do. We also find that models are far more likely to generate similar answers to template-based prompts than to template-free ones. This is in line with work that show that models often rely on simple heuristics and keywords from the data for prediction [178, 126, 94]. This is a significant issue for template-based probing, as the keywords are the same for each template (e.g., "born-in"). In comparison, template-free probes often provide a unique prompt per entity. This seems neither a result of using [125]'s probing method nor a domain-specific issue, as this can be seen in both the generic domain and also the biomedical domain using the original probing technique (i.e., not entity ranking) in [281]'s paper.

To quantify this behavior, we calculate the number of times each entity appears in each model's top-10 predictions. Figure 3.2 shows our result for the Google-RE datasets. Two obvious things appear: 1) models that score highest on the datasets, such as ALBERT base and large, BERT base and large, and DistilBERT, predict the same entities the least on both the template-free and template-based datasets; 2) converting the template-free prompts to template-based result in an increase of the amount of times each model predicts similar entities. Notably, Bio ClinicalBERT

has an increase of roughly 40% in the amount of times it predicts the top-1 entity, where Discharge BERT has an increase of roughly 30%. We can also see that the amount of unique entities each model predicts is also significantly reduced when converting to templates. For example, Albert-large goes from 0.6% to 0.24% on top-1 and Bio ClinicalBERT goes from 0.77% to 0.15% on top-10.

3.2.4.5 Pseudo-perplexity

We further measure the models' average certainty for both probing techniques. As perplexity is undefined for masked LMs like the ones we evaluate, we follow [246]'s approach to compute a model's pseudo-perplexity. We create t copies of a sentence, with t being the number of tokens in the sentence, and mask one token at a time. Then, we pass the token IDs per sentence to the models, and get the average negative log-likelihood for each token. Summing the above and taking an exponentiated average results in:

$$\text{PPL}(X) = exp\{-\frac{1}{t}\sum_{i=1}^{t}\log(p_\theta(x_i \mid x \neq i))\}$$

where $\log(p_\theta(x_i \mid x \neq i))$ is the log-likelihood of the ith token, where $[0 \leq i \leq t]$, conditioned on the remaining tokens $x \neq i$.

Results can be seen in Figure 3.3 and Table 3.13. For template-free datasets, model perplexity decreases as accuracy increases. Surprisingly, however, we see the opposite for template-based datasets. This is unexpected, as it suggests that, as models get less certain about their answers, they perform better. A potential explanation is that models that are less certain about their answers are less likely to predict similar entities for the same template.

We also find a strange behavior for the template-based datasets regarding model size. Larger models generally perform better than smaller ones. However, we find that many times the smaller models have a lower perplexity. We only find one such occurrence – the BERT models — in the template-free ConceptNet.

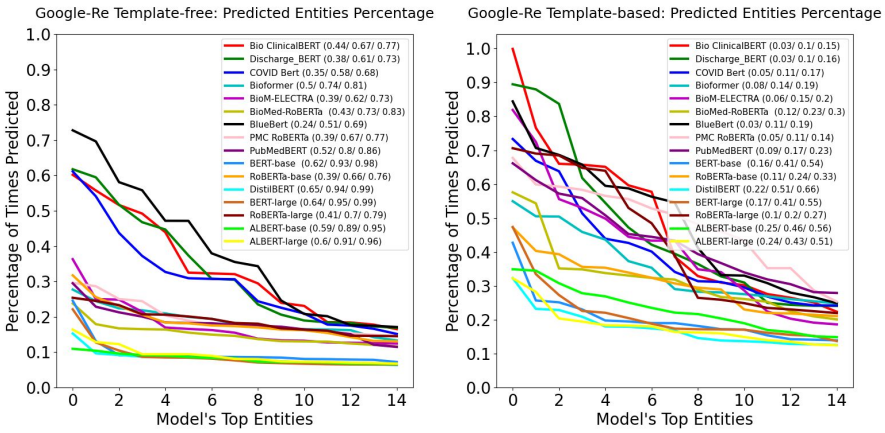


Figure 3.2: Template-free vs. template-based: We evaluate the percentage of times each entity appears in the top 10 predictions for each prompt. We show the results for the top 15 most frequent entities. Next to each model's name we also add the percentage of unique entities it predicts over all prompts for top 1, 5, and 10.

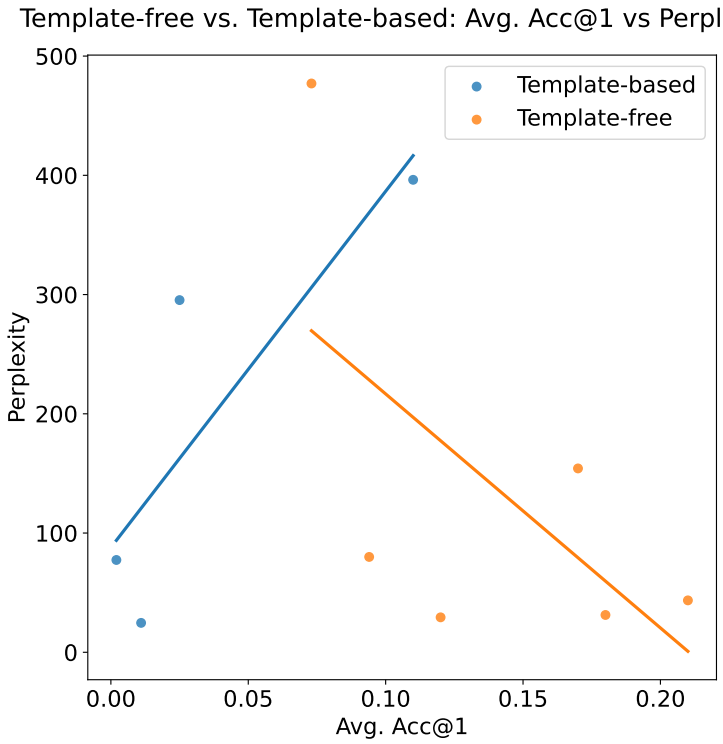


Figure 3.3: Template-free vs. Template-based: Average Acc@1 vs average Perplexity per model, over datasets. Template-based Pearson's correlation coefficient: 0.83, p-value=0.16. Template-free Pearson's correlation coefficient: 0.60, p-value=0.20.

Model	Google-RE (TB)	Google-RE (TF)	T-REx (TB)	T-REx (TF)	ConceptNet (TF)	SQuAD (TF)	LIPID-Gene (TF)	LIPID-Chem (TF)	Biomed-Wikidata (TB)	CTD (TB)
⋆PubMedBERT	76.80	23.75	255.02	46.86	129.59	90.35	10.39	13.65	10.80	50.16
⋆Bioformer	121.00	36.19	401.90	67.22	201.22	125.88	9.53	11.62	15.94	76.44
⋆BioM-ELECTRA	46.25	18.79	152.26	35.41	130.00	51.28	12.24	27.12	8.46	21.65
†BioMed-RoBERTa	172.12	25.55	335.73	20.93	343.86	73.18	8.56	10.46	18.17	26.32
†COVID Bert	84.62	25.65	212.26	43.26	247.78	88.75	5.16	5.72	5.81	11.19
†BlueBert	2345.89	719.22	2060.30	6360.65	971.47	964.89	40.15	45.83	18.67	65.21
†Discharge BERT	497.91	124.49	634.86	118.29	462.68	306.32	13.44	14.67	13.72	33.60
†PMC RoBERTa	53.79	7.68	66.05	4.87	81.76	16.05	6.32	6.62	12.04	22.13
†Bio ClinicalBERT	451.46	159.26	662.10	155.75	452.21	345.26	19.02	20.71	21.33	57.18
◇RoBERTa-base	63.99	9.82	105.80	8.11	150.24	24.34	11.26	11.62	19.05	42.09
◇RoBERTa-large	51.81	7.63	63.82	5.00	75.84	16.22	7.21	7.48	13.32	27.02
◇BERT-base	24.51	7.36	35.57	7.44	126.63	25.23	14.43	15.90	7.09	17.61
◇BERT-large	25.14	6.83	36.23	6.06	187.94	23.44	13.24	13.95	6.02	14.99
◇ALBERT-base	423.59	71.01	878.23	90.15	3720.06	163.88	197.73	193.15	13.32	365.75
◇ALBERT-large	260.91	27.51	406.03	33.37	235.64	120.33	85.15	86.55	95.03	380.45
◇DistilBERT	25.96	8.77	33.50	10.40	115.47	31.69	15.22	15.64	9.75	26.99
Average	295.36	79.97	396.23	43.54[5]	477.02	154.19	29.31	31.29	24.66	77.42

Table 3.13: **Perplexity** results. We report average perplexity for each model in addition to the average on each dataset. TF=Template-free. TB=Template-based.

3.2.4.6 Amount of Entities

While the entity ranking technique we use (see Section 3.2.2) allows us to circumvent the fact that real-world entities are often composed of more than one token, it is also susceptible to the amount of entities: as models are tasked with ranking entities, datasets with a larger number of unique entities are harder as there are more options. To quantify the effect the number of entities has on model accuracy we select three datasets – SQuAD, Biomed-Wikidata, and CTD — and increase the amount of entities as follows: for SQuAD, we add all entities from the development set (8827 unique entities in comparison to 303), for Biomed-Wikidata, we add all entities from the CTD dataset (4514 in comparison to 1267), and for the CTD dataset we add all entities from the Biomed-Wikidata dataset (4514 in comparison to 3251).

Table 3.14 shows our results. There is a performance drop across all models and datasets. While for the CTD and Wikidata datasets that drop is relatively low, on the SQuAD dataset the effect is more significant. This difference may be a result of the number of entities added to each dataset or the difficulty of the task. While the number of entities may have some effect on models scores, it is important to note that on the two parallel datasets – Google-RE and T-REx, the number of entities stays the same and the only different variable is the probing method. However,

Model	SQuAD Original			SQuAD Mod.			Biomed-Wikidata Original			Biomed-Wiki Mod.			CTD Original			CTD Mod.		
	A@1	A@5	A@10	A@1	A@5	A@10	A@1	A@5	A@10	A@1	A@5	A@10	A@1	A@5	A@10	A@1	A@5	A@10
⋆PubMedBERT	$1.8e^{-1}$	$3.7e^{-1}$	$4.5e^{-1}$	$9.9e^{-2}$	$2.2e^{-1}$	$2.7e^{-1}$	$4.4e^{-2}$	$1.3e^{-1}$	$1.9e^{-1}$	$4.2e^{-2}$	$1.0e^{-1}$	$1.6e^{-1}$	$7.8e^{-3}$	$2.9e^{-2}$	$4.5e^{-2}$	$7.3e^{-3}$	$2.2e^{-2}$	$3.7e^{-2}$
⋆Bioformer	$1.2e^{-1}$	$2.7e^{-1}$	$3.4e^{-1}$	$6.2e^{-2}$	$1.4e^{-1}$	$1.7e^{-1}$	$3.7e^{-2}$	$1.0e^{-1}$	$1.6e^{-1}$	$3.7e^{-2}$	$1.0e^{-1}$	$1.6e^{-1}$	$6.0e^{-3}$	$2.3e^{-2}$	$3.6e^{-2}$	$3.4e^{-3}$	$1.6e^{-2}$	$2.7e^{-2}$
⋆BioM-ELECTRA	$1.0e^{-1}$	$2.6e^{-1}$	$3.2e^{-1}$	$3.9e^{-2}$	$8.2e^{-2}$	$1.2e^{-1}$	$3.0e^{-2}$	$1.0e^{-1}$	$1.5e^{-1}$	$3.0e^{-2}$	$8.3e^{-2}$	$1.2e^{-1}$	$3.4e^{-3}$	$1.4e^{-2}$	$2.6e^{-2}$	$2.9e^{-3}$	$1.1e^{-2}$	$1.9e^{-2}$
†BioMed-RoBERTa	$1.2e^{-1}$	$2.6e^{-1}$	$3.3e^{-1}$	$6.6e^{-3}$	$9.9e^{-3}$	$9.9e^{-3}$	$2.3e^{-3}$	$1.4e^{-2}$	$3.4e^{-2}$	$2.3e^{-3}$	$1.4e^{-2}$	$3.4e^{-2}$	$3.6e^{-4}$	$3.7e^{-3}$	$8.4e^{-3}$	$3.6e^{-4}$	$1.7e^{-3}$	$3.9e^{-3}$
†COVID Bert	$8.2e^{-2}$	$2.2e^{-1}$	$2.7e^{-1}$	$3.6e^{-2}$	$7.6e^{-2}$	$1.0e^{-1}$	$8.1e^{-3}$	$4.3e^{-2}$	$6.4e^{-2}$	$7.1e^{-3}$	$3.2e^{-2}$	$4.4e^{-2}$	$4.0e^{-3}$	$1.0e^{-2}$	$2.0e^{-2}$	$2.5e^{-3}$	$8.8e^{-3}$	$1.4e^{-2}$
†BlueBert	$4.9e^{-2}$	$1.5e^{-1}$	$2.2e^{-1}$	$2.3e^{-2}$	$3.3e^{-2}$	$4.3e^{-2}$	$1.3e^{-2}$	$5.1e^{-2}$	$8.7e^{-2}$	$1.2e^{-2}$	$4.1e^{-2}$	$6.6e^{-2}$	$5.4e^{-4}$	$5.4e^{-3}$	$9.8e^{-3}$	$3.6e^{-4}$	$2.8e^{-3}$	$5.5e^{-3}$
†Discharge BERT	$5.9e^{-2}$	$1.6e^{-1}$	$2.2e^{-1}$	$2.3e^{-2}$	$4.6e^{-2}$	$4.6e^{-2}$	$6.8e^{-3}$	$3.7e^{-2}$	$6.0e^{-2}$	$6.8e^{-3}$	$2.9e^{-2}$	$4.8e^{-2}$	$3.9e^{-3}$	$1.2e^{-2}$	$2.0e^{-2}$	$2.4e^{-3}$	$4.9e^{-3}$	$1.0e^{-2}$
†PMC RoBERTa	$1.1e^{-1}$	$2.7e^{-1}$	$3.5e^{-1}$	0.0	$6.6e^{-3}$	$9.9e^{-3}$	$2.0e^{-3}$	$1.7e^{-2}$	$3.0e^{-2}$	$2.0e^{-3}$	$1.6e^{-2}$	$2.9e^{-2}$	$8.1e^{-4}$	$3.7e^{-3}$	$8.5e^{-3}$	$4.5e^{-4}$	$2.5e^{-3}$	$4.5e^{-3}$
†Bio ClinicalBERT	$3.6e^{-2}$	$1.1e^{-1}$	$1.8e^{-1}$	$9.9e^{-3}$	$2.9e^{-2}$	$4.6e^{-2}$	$7.8e^{-3}$	$3.6e^{-2}$	$6.2e^{-2}$	$7.8e^{-3}$	$2.6e^{-2}$	$4.5e^{-2}$	$1.9e^{-3}$	$1.0e^{-2}$	$1.5e^{-2}$	$1.8e^{-3}$	$7.7e^{-3}$	$1.0e^{-2}$
◇RoBERTa-base	$1.0e^{-1}$	$2.0e^{-1}$	$3.0e^{-1}$	0.0	0.0	0.0	$1.3e^{-3}$	$2.2e^{-2}$	$3.6e^{-2}$	$1.3e^{-3}$	$1.9e^{-2}$	$3.1e^{-2}$	$3.6e^{-4}$	$4.4e^{-3}$	$7.9e^{-3}$	$3.6e^{-4}$	$3.3e^{-3}$	$5.8e^{-3}$
◇RoBERTa-large	$1.4e^{-1}$	$3.0e^{-1}$	$4.0e^{-1}$	0.0	$6.6e^{-3}$	$6.6e^{-3}$	$2.0e^{-3}$	$2.0e^{-2}$	$3.7e^{-2}$	$2.0e^{-3}$	$1.9e^{-2}$	$3.5e^{-2}$	$5.4e^{-4}$	$3.7e^{-3}$	$7.4e^{-3}$	0.0	$1.6e^{-3}$	$3.7e^{-3}$
◇BERT-base	$3.6e^{-1}$	$6.9e^{-1}$	$8.1e^{-1}$	$2.2e^{-1}$	$4.2e^{-1}$	$5.0e^{-1}$	$7.8e^{-3}$	$3.4e^{-2}$	$6.0e^{-2}$	$7.5e^{-3}$	$2.3e^{-2}$	$4.0e^{-2}$	$8.1e^{-4}$	$3.9e^{-3}$	$9.6e^{-3}$	0.0	$1.5e^{-3}$	$5.0e^{-3}$
◇BERT-large	$4.4e^{-1}$	$7.7e^{-1}$	$8.6e^{-1}$	$2.7e^{-1}$	$5.3e^{-1}$	$6.1e^{-1}$	$7.5e^{-3}$	$3.9e^{-2}$	$5.2e^{-2}$	$6.8e^{-3}$	$2.6e^{-2}$	$4.0e^{-2}$	$1.3e^{-3}$	$5.3e^{-3}$	$1.\ e^{-2}$	$1.8e^{-4}$	$2.3e^{-3}$	$6.0e^{-3}$
◇ALBERT-base	$2.5e^{-1}$	$5.3e^{-1}$	$6.2e^{-1}$	$1.6e^{-1}$	$3.6e^{-1}$	$4.5e^{-1}$	$3.0e^{-3}$	$3.3e^{-2}$	$5.6e^{-2}$	$2.7e^{-3}$	$1.8e^{-2}$	$3.2e^{-2}$	$4.5e^{-4}$	$3.8e^{-3}$	$1.0e^{-2}$	$3.6e^{-4}$	$1.0e^{-3}$	$5.\ e^{-3}$
◇ALBERT-large	$3.0e^{-1}$	$5.8e^{-1}$	$7.1e^{-1}$	$2.1e^{-1}$	$4.1e^{-1}$	$5.0e^{-1}$	$1.0e^{-3}$	$1.9e^{-2}$	$3.5e^{-2}$	$1.0e^{-3}$	$1.0e^{-3}$	$2.1e^{-2}$	$4.5e^{-4}$	$2.2e^{-3}$	$6.6e^{-3}$	$1.8e^{-4}$	$1.6e^{-3}$	$2.9e^{-3}$
◇DistilBERT	$3.8e^{-1}$	$7.3e^{-1}$	$8.2e^{-1}$	$2.4e^{-1}$	$4.9e^{-1}$	$5.6e^{-1}$	$1.1e^{-2}$	$4.2e^{-2}$	$6.8e^{-2}$	$1.0e^{-2}$	$2.7e^{-2}$	$5.4e^{-2}$	$9.0e^{-4}$	$4.7e^{-3}$	$1.1e^{-2}$	$8.1e^{-4}$	$2.2e^{-3}$	$4.7e^{-3}$
Average	$1.7e^{-1}$	$3.6e^{-1}$	$4.5e^{-1}$	$8.7e^{-2}$	$1.7e^{-1}$	$2.1e^{-1}$	$1.1e^{-2}$	$4.6e^{-2}$	$7.3e^{-2}$	$1.1e^{-2}$	$3.5e^{-2}$	$5.9e^{-2}$	$2.0e^{-3}$	$8.6e^{-3}$	$1.5e^{-2}$	$1.4e^{-3}$	$5.6e^{-3}$	$1.0e^{-2}$

Table 3.14: Accuracy ($A@k$) of our models for different numbers of entities.

the ranking still changes.

3.2.5 Which Method Should We Use?

The obvious question is: *Which probing technique should we use: template-based or template-free?*

As described in [219], the cloze-task measures the lower bound for what LMs know. From that regard, we find that the template-free approach results in a higher lower bound of knowledge, and hence, we conclude that a better method to evaluate the *amount of model knowledge* is the template-free approach.

From a cost perspective, it is also much cheaper to develop template-free datasets, as they do not require domain experts.

Lastly, our analyses suggest that the two techniques may evaluate *different kinds of knowledge*. It is, e.g., unclear why smaller models often have better performance (e.g., on perplexity or Acc@K) than their larger counterparts in the template-based approach, but almost always lower performance using the template-free approach. This suggest that it is best to use multiple probing methods to assess the factual information these models contain.

Chapter 4

Knowledge Usage

In parallel with understanding how to determine the extent of the knowledge in a model, I carry out research to characterize how a model uses internal and external knowledge when answering questions.

4.1 It Is Not About What You Say, It Is About How You Say It: A Surprisingly Simple Approach for Improving Reading Comprehension

The work described in this section has been published in Findings of ACL 2024 [258].

4.1.1 Introduction

For the task of reading comprehension (RC), models receive two kinds of inputs: 1) a context, e.g., a Wikipedia article, and 2) a question that should be answered according to the context [64, 335]. While early efforts to address this task usually involve models that encode each of these separately [341, 284, 203, 46, 40], more recently, LLMs receive a concatenation of the two inputs [309, 103, 280, 15, 14, 29, 41, 44].

Surprisingly, **there is no current standard of what the ordering of such input components should be**. For example, [280, 204, 15, 120, 276, 344] provide the question first in each prompt, while [37, 204, 156, 14, 29, 276, 41, 44] provide the context first. Moreover, **there is no current standard of how to present the two input components in general**. For example,

considering the question and context strings $< q >$ and $< c >$, respectively, [309] add the special tokens "question:" and "context:" before the question and context, while [204] use "$< c >$**Question:** $< q >$", [344] use "[Question]: $< q >$ [Passage]: $< c >$", [156] use "$< c > < q >$", and others such as [14, 29, 41, 44], employ their own methods.

While at first sight this might not seem important, many works have shown that LMs can be extremely susceptible to slight variations in the input sequence [109, 273, 253, 255]. Furthermore, recent research has found that **different presentations of inputs can help emphasize them** and improve models' ability to follow instructions [340]. Based on these observations, we ask the following research questions (RQs): 1) How does the order of inputs – i.e., question and context – affect model performance? 2) Does emphasizing either the question, the context, or both enhance performance? A summary of these questions can be seen in Figure 5.2.

We evaluate 9 LLMs on 3 datasets and find the following: 1) The ordering of the question and context is crucial, and improves model performance with an accuracy increase of up to 31%. 2) Both prompt-based and attention-based emphasis methods are capable of strongly improving models' performance, where emphasizing the context yields superior results compared to emphasizing the question, and in general, emphasizing parts of the input is particularly effective for addressing questions that models lack the parametric knowledge to answer. 3) The best emphasis method is surprisingly simple: it only requires a simple concatenation of a few tokens to the input and results in an accuracy improvement of up to 36%, allowing smaller models to outperform their significantly larger counterparts.

4.1.2 Models

We experiment with 9 different LLMs.

Llama-2-7B and Llama-2-13B Llama-2-7B and Llama-2-13B [290] are LLMs which contain 7 and 13 billion parameters, respectively, and are trained on 2 trillion tokens. We use these models as they perform well on the reading comprehension task [290] and recent work shows that their performance can be improved using emphasis methods [340].

Falcon-7B and Falcon-7B Instruct These two models contain 7 billion parameters each, and are trained on 1.5 trillion tokens [5]. We opt for these models because they are newer and have demonstrated significant success across various tasks. Additionally, Falcon-7B Instruct comes with an instruct version, enabling us to compare the performance of both variations.

MPT-7B and MPT-7B Instruct These are two LLMs with 7 billion parameters, trained on 1 trillion tokens. Chosen for their recent development and proven versatility.

GPT-J-6B GPT-J-6B [298] contains 6 billion parameters and is trained on the Pile dataset [79]. We use this model as in addition to the fact that it has been shown to perform well on question answering tasks [51], it is also often compared again our largest model – Llama-2 [290, 340] and recent work shows that its performance can be improved using emphasis methods [340].

GPT-2-XL GPT-2-XL [226] a LLM with 1.5 billion parameters and is trained on WebText [226]. While much smaller than current state-of-the-art models, such as ChatGPT or GPT 4 [207], we experiment with it as many low-resource settings require usage of smaller models.

GPT-2-Large Our last model, GPT-2-Large [226], contains 774 million parameters and, similar to GPT-2-XL, is trained on WebText. We use it for similar reasons as those we described in the GPT-2-XL Section.

4.1.3 Experiments

4.1.3.1 Datasets

We experiment with the following RC datasets:

Natural Questions The natural questions dataset [138] is comprised of authentic, anonymized, and aggregated queries directed to the Google search engine. Each question is accompanied by an entire Wikipedia page, and a collection of annotated long and short answers. As entire Wikipedia pages exceed many of our models' context lengths, for each question, we use each of the long answers as the context and the corresponding short answers as the gold answers.

We utilize it due to its widespread adoption and popularity within the research community,

ensuring the reproducibility and comparability of our results with existing studies. Additionally, its comprehensive coverage of diverse question types and real-world contexts allows us to further evaluate whether our findings generalize.

Stanford Question Answering Dataset (SQuAD) SQuAD [233] is composed of questions that are gathered from crowdworkers who ask questions about Wikipedia articles. We choose to use it for similar reasons described as the Natural Questions dataset.[1]

AdversarialQA The AdversarialQA dataset [19] has been constructed adversarially, based on 3 models-in-the-loop. More specifically, the authors use the same SQuAD annotation methodology and models trained on it, and explore an annotation setting where annotators are tasked with formulating questions for which the model yields incorrect predictions. Consequently, the dataset is composed solely of instances where models answer inaccurately. While not as popular as SQuAD or the Natural Questions, we utilize this dataset as this annotation methodology makes these questions unique and especially challenging.

Data Splits As the test set for each of these datasets is either private or does not contain gold answers, we randomly split the validation sets into two parts and use one half as our validation set and the other as our held-out test set. This results in roughly the following split for each dataset. Natural Questions: $307k$ train, 3915 validation, 3915 test, SQuAD: $87k$ train, 5285 validation, 5285 test, AdversarialQA: $30k$ train, 1500 validation, 1500 test.

4.1.3.2 Prompt Structure

RC datasets consist of question, context, and answer triples (q, c, a), where $q \in Q$, $c \in C$, $a \in A$. As outlined above, our RQ1 is concerned with the order in which the question and context are provided to the model: since previous work has been inconsistent in this regard, we explore which order (if any) results in higher performance.

Concretely, we compare the following two prompt structures (cf. Figure 5.2):

[1] We use the 1.0 version instead of the 2.0 version, as the later version contains empty strings as labels for its irrelevant contexts, which prevents us from using the closed-book setting to determine its parametric knowledge (see Section 4.1.3.5).

Question First Here, the question comes first in the prompt. In our concrete format, this results in the input sequence

$$\texttt{Question: } < q > \texttt{ Context: } < c >,$$

where q and c are pairs of question and context strings, $q \in Q$, $c \in C$.

Context First In this setting, the context is the first part of the prompt. In our concrete format, this results in the input sequence

$$\texttt{Context: } < c > \texttt{ Question: } < q >,$$

where, again, q and c are question–context pairs, $q \in Q$, $c \in C$.

4.1.3.3 Emphasis Strategies

Marked Prompting MP [340] is a simple prompt-based approach in which we append a string to the input sequence in order to emphasize it. For example, to emphasize the questions, we can append the string **“ * ”** to

$$\text{Question: } < q > \text{ Context: } < c >$$

which would result in

$$\text{Question: } *< q >* \text{ Context: } < c >$$

We experiment with 4 MP methods, composed of the following start and end string pairs: [* and *, “ and ”, <emphasize> and <\emphasize>, <mark> and <\mark>]

Attention Steering In comparison to MP, AS is a more computationally-intensive method to emphasize input tokens and is attention-based.

We follow [340]'s approach known as PASTA, which requires 1) an LLM with L stacked layers, each with N multi-head attention (MHA) submodules, such as most transformer-based models [294]; 2) input text W, and 3) a segment $w \in W$ that is found within the input text.

PASTA is composed of two parts:

1) *Attention steering*: in this part, we downweight the attention scores of any token that is not part of the segment w, by multiplying them with a small scalar $0 \leq \alpha < 1$ for a selected $n \in N$ MHA submodules. In our experiments, we use $\alpha = 1e^{-3}$ based on [340].

2) *Model profiling*: here, we select which $n \in N$ to apply the AS to. While the original paper experiments with several selection methods, such as applying the steering to all heads, single heads, or entire layers, they obtain the best performance when selecting the intersection of the top-k best performing heads across several datasets. They select k from a small number of options, such as $\{300, 400, 500\}$ for Llama 7B. However, we find that we can improve performance by increasing this range.

In particular, from each dataset's *training split* D_{ti}, we take a small subset of examples $d_{ti} \in D_{ti}$, and apply AS to each head individually. In our experiments, we use $|d_{ti}| = 1000$ for GPT-2 large and XL, and $|d_{ti}| = 500$ for GPT-J and Llama-2, for computational reasons, after manually assessing different values which result in roughly similar models' scores. We store the performance of the model for each head, which results in $L * N$ scores for each d_{ti}. Next, on each dataset's *validation split* D_{vi} we iteratively select a k, where $0 < k \leq N * L$, and find the intersection of the top-k performing heads across all datasets $d_{ti} \in D_{ti}$. We store the scores, which results in $L * N$ scores for each k for each D_{vi}. For the test split, we use the best k based on the validation split.

Baseline: No Emphasis As a baseline, we further compare to a setting in which we do not emphasize any string and use the original prompt from Section 4.1.3.2 as inputs to the models.

4.1.3.4 Hyperparameters

We use a maximum sequence length of 512. Truncation due to this might result in an unfair comparison between the different prompt structures as either question or context might get truncated.[2]

In order to avoid this, we remove sequences that are longer than 512 tokens (about 15% of the

[2] See Section 4.1.6.4 for an analysis of models with a larger context length.

examples in the Natural Questions dataset, less than 1% for SQuAD, and 0% for AdversarialQA).

4.1.3.5 Metrics

Accuracy Following [157, 121, 171], we assess the performance of all models using accuracy, determining if any of the gold responses are present in the predicted output. Concretely, we feed the two prompts described in Section 4.1.3.2, such as *"Question:* $< q >$*. Context:* $< c >$*"*, to each of the models, and evaluate whether the gold label answer exists within the LLM generated answer.[3]

Context-free Accuracy We are further interested in evaluating the models' parametric knowledge. For this, we follow work by [259, 150, 316, 240], who use a closed-book setting to evaluate models' parametric knowledge. In particular, we define *known knowledge* as questions that models answers correctly without the corresponding context and *unknown knowledge* as those they cannot.

Perplexity Perplexity (PPL) is defined as the exponentiated average of the negative log-likelihood of a sequence. Concretely, given a sequence of tokens $X = (x_0, x_1, ..., x_t)$, the perplexity of X denoted as

$$PPL(X) = exp(-\frac{1}{t}\sum_i^t logp_\theta(x_i \mid x_{<i}))$$

where $logp_\theta(x_i \mid x_{<i})$ represents the log-likelihood of the i-th token conditioned on the preceding tokens $x_{<i}$ according to the model.

4.1.4 Results

4.1.4.1 RQ 1: Question First vs. Context First

We first analyze whether models' performance differs when given the same information, but in different order: question-first and context-first. Our results can be seen in Table 4.1.

[3] While this approach is popular, it is important to note that no existing evaluation metric is flawless. For instance, this approach may overlook accurate responses (e.g., because they are not an exact match to gold answers) or erroneously categorize incorrect responses as correct. To address this concern, we supplement our evaluation process by manually inspecting 100 responses from Llama 2 on the Natural Questions dataset in the no emphasis, context-first setting, to evaluate the frequency of such occurrences. We find that while this approach identifies 58.1% of the answers as correct, manual analysis identifies 82%. This highlights the gap between this popular method and human evaluation.

No Emphasis Accuracy As we aim to understand the effect that prompt structure alone has on models' performance, for this analysis we focus on the no emphasis (NE) baseline.

Looking at the NE setting, there is a clear difference across almost all models and datasets. More specifically, prompting models with the context first strongly improves performance, with an average increase of 13.46% (49.90% in comparison to 36.44%). On the Natural Questions dataset, the highest accuracy change occurs for GPT-J: from 33.3% to 64.5% (31.2% difference). The second highest change is seen for GPT-2-XL: from 28.0% to 51.2% (23.2% difference). The third highest change occurs for Llama-2, which scores 46.3% when the question is given first but 58.1% when the context is given first (11.8% difference). Similar behavior can be seen for the SQuAD and AdversarialQA datasets as well. For example, Llama-2 changes from 60.4% to 72.9% on SQuAD, and GPT-2-XL changes from 24.8% to 31.8% on AdversarialQA. However, we do find two cases where placing the context first does not improve the results, and actually slightly reduces them: on the AdversarialQA dataset, GPT-2 large and GPT-J change from 27.7% to 26.9% and 47.2% to 46.2%, respectively.

4.1.4.2 RQ 2: Emphasis and Performance

We next analyze whether emphasizing parts of the input – the question, the context, or both – enhances models' performance. Our results can be seen again in Table 4.1.

Performance Improvement Across Almost All Settings We find that across all datasets, models, and prompt structures, there is a performance difference between emphasizing either the context, the question, or both, which will further be discussed in Section 4.1.6.1. However, emphasizing parts of the input is overall beneficial and can strongly improve models' NE performance. For example, on the Natural Questions dataset, every emphasis method improves Llama-2 NE performance for the question-first setting (except for emphasizing the context using MP-*). To more concretely assess the overall performance improvement emphasizing the input entails, we compare the averaged NE performance across all models, dataset, and settings, to the averaged performance over all emphasis methods, models, datasets, and settings. We find that,

while the average NE performance is 43.17%, the average model performance when emphasizing the input is 47.31%.

4.1.5 Analysis and Discussion

4.1.6 Sequence Order Analysis

No-emphasis Perplexity To further understand the behavior we find from our analysis of RQ1 in Section 4.1.4.1, we evaluate the average perplexity of the prompts under each model for each of the two prompt structures – *Question:* $< q >$ *Context:* $< c >$ and *Context:* $< c >$ *Question:* $< q >$ –, each dataset and the NE setting. Our results can be seen in Table 4.2.

Across almost all dataset, models' perplexity is lower (i.e., "better") for the context-first setting, with an average reduction of 1.77 on the Natural Questions (25.48 vs. 23.70), 1.77 on SQuAD (16.12 vs. 15.94), 0.24 on AdversarialQA (17.82 vs. 17.57), and over all datasets of 0.73 (19.81 vs. 19.07). For example, the highest perplexity reduction occurs for GPT-2 large, which scores 32.22 on the Natural Questions dataset when the context is provided first, in comparison to 36.26 for the question-first setting (4.04 difference).

Perplexity vs. Accuracy Surprisingly, looking at Table 4.2 for the two cases above in which placing the context first does not improve accuracy (GPT-2 large and GPT-J on AdversarialQA), we find that only GPT-2 large scores higher on perplexity for the context-first setting, which could potentially explain the accuracy difference as the model finds this prompt structure more confusing on this particular dataset. However, we do not find that the perplexity was higher for the questions-first structure for GPT-J. Moreover, we find two more cases where models' perplexity was higher for one of the structures, but accuracy was higher on the same structure: Llama-2 on Natural Questions and GPT-2 large on SQuAD. This suggests that while the models do not find the context-first structure more confusing (as measured by their perplexity), they score lower on accuracy for another reason.

4.1.6.1 Emphasis Analysis

Different Emphasis Methods Affect Similar Models Differently We find that different emphasis methods affect similar models differently. On the Natural Questions dataset, while emphasizing the context using the MP-<emphasize> method on GPT-J on the question-first structure increases its NE accuracy from 33.3% to 69.0%, outperforming all other models, using the MP-* method reduces its score to 26.9%.

Similar Emphasis Methods Affect Different Models Differently We also find that similar emphasis methods affect different models differently. For example, on the AdversarialQA dataset and the context-first, context-emphasis setting, AS improves Llama-2 NE performance from 49.4% to 53.3%, and GPT-2-XL's NE performance from 26.9% to 33.6%. However, AS reduces GPT-J's performance from 46.2% to 41.7%.

Best Emphasis Methods To assess which emphasis methods are best for each model, we average the scores across all datasets and settings for each model. We find that the top 3 best emphasis methods for each model are (in decreasing order): Llama-2: (MP-", MP-<mark>, MP-<emphasize>), GPT-J: (MP-<emphasize>, MP-<mark>, MP-"), GPT-2 large: (AS, MP-<emphasize>, MP-<mark>), and GPT-2-XL: (AS, MP-<mark>, MP-<emphasize>).

Overall, across all models, datasets and settings, the best emphasis method may seem to be AS, with an average accuracy of 49.39%. This is aligned with [340]'s result, which finds that AS outperforms two MP methods on the task of instruction following.

However, looking at the top accuracies for each model on each dataset, we actually find that AS only outperforms other emphasis methods 6 out of the 24 times (4 models, 2 prompt structures for each, on 3 datasets). And from that regard, MP outperforms it (MP also scores fairly close to it overall, with the highest average accuracy of 48.68% for MP-<emphasize>).

Emphasis on C vs. Q vs. CQ To analyze which substring is better to emphasize – the context, the question, or both –, we average the performance of all models across all datasets, emphasis methods, and prompt structures. We find that the highest performance is achieved by

emphasizing both context and question, with an average accuracy score of 49.49%. However, we also find that emphasizing the context is roughly just as good, with an average accuracy score of 49.21%, and that emphasizing the question falls much below both, with an average accuracy score of 43.68%.

Does Size Matter? Here, we analyze whether models' size affects their ability to be emphasized by looking at the best method for each on each setting. And while we do not find a clear pattern, we find some cases that suggest that emphasis methods are more beneficial for smaller models. For example, on the SQuAD dataset and the question-first setting, GPT-2 large improves from 27.1% to 56.5% using the MP-<mark> method (29.4% improvement), where GPT-J improves from 45.5% to 64.7% using the MP-<emphasis> method (19.2% improvement), and Llama-2 from 60.4% to 72.3% using the MP-" method (11.9% improvement).

Does Training Data Matter? To evaluate the effect training data has on the susceptibility of models for being emphasized, we compare GPT-2 large and GPT-2-XL as they are trained on the same corpus. From Table 4.1 we can see that, while these two models are trained on similar data, on many occasions, similar emphasis methods result in different behavior. For example, on the question-first setting and the Natural Questions dataset, while AS result in the highest performance when applied to the question on both models, for context emphasis, the best method for GPT-2 large is AS, where for GPT-2-XL the best method is MP-<mark> or MP-<emphasize>. We also do not find the same absolute improvements across the two models when looking at similar emphasis methods and similar settings. This suggests that, while the training data has some effect on which emphasis method is beneficial for each model, it is not the whole story.

Attention Heads Analysis

To further understand why different emphasis methods result in different models' scores we evaluate the attention scores for the strings that are being emphasized by the different methods on the question-first setting. More concretely, for each MP method, we send each sentence from the Natural Questions dataset to the model. We then average the attention scores across all model's heads and layers for the tokens corresponding to the string to be emphasized – either the context

or the question. Our results can be seen in Table 4.3.

We do not find a clear pattern that highlights whether emphasis methods result in a higher or lower attention scores for emphasis strings. For example, while GPT 2 large has an increase of accuracy from 22.1% to 29.7% when changing from the MP-* method to the MP-" method on the question-emphasis setting, the attention scores stay the same. We also see that sometimes the attention scores go up when accuracy go down, such as in GPT 2 XL, MP-mark to MP-* on question emphasis, and sometimes the attention scores go down when accuracy go up, such as in GPT 2 large, MP-" to MP-emphasis, on the context emphasis setting.

4.1.6.2 Known Vs. Unknown Knowledge

Marked Prompting We next evaluate whether MP, and specifically the best performing setting overall – context-first, question + context emphasis –, works better for addressing knowledge that models have or do not have. Our results can be seen in Table 4.4.

We can see that, across almost all three datasets and all models, emphasizing the input string on the unknown knowledge split results in more improvement than emphasizing the input string on the known knowledge split. For example, on Natural Questions, for unknown knowledge, Llama-2 and GPT-J improve from 46.4% and 63.2% to 49.9% and 65.5%, respectively. Where on the known knowledge split, they respectively change from 93.4% to 93.6% and from 88.5% to 85.2%.

One potential explanation for that is that models tend to already perform reasonably well on known knowledge, since they have most likely acquired that knowledge during training. However, emphasizing input strings on unknown knowledge forces the model to adapt its learned representations to handle unseen or less familiar data.

Attention Steering Next, we evaluate whether AS, and specifically the best performing setting of AS – question-first, question steering –, works better for addressing knowledge that models have or do not have. Our results can be seen in Table 4.5.

Across almost all three datasets and all models, steering the input string in the unknown knowledge split results in more improvement than steering it in the known knowledge split. For

example, on Natural Questions, for unknown knowledge, GPT-J and GPT-2 Large improve from 27.9% and 29.4% to 59.9% and 54.6%, respectively. In contrast, on the known knowledge split, they improve from 56.4% to 76.8% and from 52.1% to 71.4%, respectively.

4.1.6.3 Can Emphasis Be Bad?

While we find that emphasizing parts of the input using various emphasis methods can be beneficial, it does require experimentation, as choosing the wrong emphasis method can actually be disadvantageous. Averaging over all datasets, models, and settings in Table 4.1, we find that the worse emphasis method is MP-*, only increasing the average accuracy from 43.17% to 43.66%, and at its worst setting it reduces Llama-2's baseline performance from 46.3% to 31.6% on the Natural Questions dataset in the question-first setting.

4.1.6.4 Newer Models, Instruction Tuning, and Max Context Length

In addition to our main results, we also add an analysis of five more LLMs, all of which were published in 2023 or afterwards and contain between 7B and 13B parameters. Two of the five additional LLMs were instruction-tuned, to evaluate whether such tuning affect the performance change due to different emphasis methods. Lastly, all five of the additional models were evaluated using their maximum context size (up to 4k). Our results can be seen in Table 4.6.

Notably, 1) Our results still hold: A) the ordering of inputs plays a crucial role in all models' performances, where putting the context first strongly improves performance; B) emphasis methods also improve models' performances. 2) The context size does not play a role in the results, in the sense that our initial results and conclusions still hold. 3) Instruction-tuned models are also susceptible to input order and emphasis methods.

4.2 Desiderata For The Context Use Of Question Answering Systems

The work described in this section has been published in EACL 2024 [259].

4.2.1 Introduction

QA systems which are based on LLMs play a larger role than ever before in our society, due to their ability to offer quick access to information [219, 240, 269, 281, 114]. Many QA systems can make use of context information when available, which often contains relevant information to help systems answer questions, cf. Figure 5.2.

We refer to all systems that are able to leverage such context information as *context-based QA systems*.

Many aspects of such systems have been evaluated by previous work, such as the amount of their parametric knowledge [219] and their robustness to noise [109], conflicting knowledge [210, 163], or irrelevant contexts [150, 199] . However, looking at such aspects in isolation makes it difficult to see trends across problems, e.g., to explore whether there is a connection between a model's attention to context and its ability to handle noise.

Here, we 1) outline a set of – previously discussed as well as novel – desiderata for context-based QA models and 2) provide an extensive survey of related works, which we group and discuss according to our desiderata. Such desiderata unify some of the existing aspects from the literature, e.g., robustness to conflicting knowledge, and outline how a QA system should behave from the perspective of the context. We will publicly release a toolkit to prepare datasets — both free-form and multiple choice (MC) type – to evaluate models according to all desiderata *at once*.

Using our toolkit, we 3) evaluate 15 LLM-based QA systems and first confirm prior works' results: while some systems appear nearly perfect, scoring 99% accuracy on standard datasets, their performance is significantly worse according to many of our desiderata. For instance, their accuracy drops by up to 93% with noise, such as random strings as distractors. Second, considering all desiderata *at once*, we find that (1) systems that are less susceptible to noise are more consistent with their answers when provided with irrelevant context; (2) most systems that are more susceptible to noise are more likely to correctly answer according to a context that conflicts with their parametric knowledge; and (3) the combination of conflicting knowledge and noise can

reduce system performance by up to 96%. Finding these novel trends using our desiderata opens new avenues to improve QA models.

4.2.2 Desiderata

We now develop a set of desiderata regarding the context use of a model, before presenting our survey on what prior work has found with regards to our desiderata and performing our own experiments in the next two sections. To come up with our desiderata, which are presented in Table 4.7, we consider the question: *How would an ideal QA model behave for different types of context?*

The ideal behavior depends on whether the knowledge in the context is known or unknown to the model. For example, looking at Table 4.7, Row 6, while systems are expected to predict the true answer for known knowledge, as they contain the relevant context within their parameters, the ideal system would answer incorrectly/"unanswerable" for unknown knowledge given irrelevant context.

We follow the work by [150, 316, 240] and define known knowledge as questions that a model can answer correctly without context, and unknown knowledge as those it cannot.

Proposed Desiderata An ideal model should:

■ For both known and unknown knowledge:

a. *Answer correctly with the original context*: this is the standard QA systems evaluation approach.

b. *Answer correctly with a noisy irrelevant variation of the original context*: QA systems should be robust to distractors, as different users and information retrieval (IR) systems introduce varying amounts of irrelevant information.

c. *Change its answer with conflicting context to the conflicting knowledge*: As our world is constantly changing, QA systems should be dynamic in their knowledge. That is, similarly to [349, 150], we believe that the context should *always* take priority over a model's parametric knowledge, when relevant.

■ For known knowledge:

d. *Answer correctly with no context*: In our setting this happens by default for known knowledge,

as by definition known knowledge is questions that can be answered without context. However, we expect the ideal system to have the largest possible amount of knowledge, i.e., to be able to answer most questions without context.

e. *Answer correctly with an irrelevant context*: Since the model answers questions correctly without context for known knowledge, it should also answer correctly with irrelevant context.

■ For unknown knowledge:

f. *Answer incorrectly/"unanswerable" with no context*: In our setting this happens by default for unknown knowledge.

While the ideal model should predict "unanswerable" for questions it cannot answer, most existing datasets do not include questions that, according to our definition, are truly "unanswerable," as they *can* be answered with parametric knowledge (cf. Sec. 4.2.4.3). Hence, we add here that models may also predict an incorrect answer, as expected from models that are forced to predict any answer other than "unanswerable" for unknown knowledge.

g. *Answer the same with irrelevant context as with no context*:

The ideal model should be consistent in its answer, even when wrong. Hence, the model's answer with irrelevant context and no context – (f) above – should be the same.

4.2.3 Survey of Prior Work

4.2.3.1 Known vs. Unknown Knowledge

As mentioned in Sec. 4.2.2, the ideal behavior depends on if the knowledge contained in the context is known or unknown to the model. While most work evaluate on the entire data without such distinction, some analyze the known knowledge split: [316, 150] analyze models using a closed-book setting, [199] assume the original contexts are known knowledge, and [34] evaluate correctly answered questions.

4.2.3.2 The Standard Approach

Row 1 in Table 4.7 shows the standard approach for evaluating QA systems, where systems are tasked with answering questions using a fixed context. For lack of space and since the focus of our survey is not the standard approach, we refer interesting readers to [336] and [64] for further reading.

4.2.3.3 Context + Distractor

Next, we focus on Row 2 in Table 4.7: the original context with a distractor, which measures the robustness of systems to various types of irrelevant (but not conflicting) noise.

Overview Many analyze the susceptibility of QA systems to context-based noise. [109] propose adding sentences that look similar to questions or random distractor words, which result in over 50% decrease in performance. However, [301] mention that such unnatural distractors allow models to easily distinguish them and ignore them. Instead, they modify their approach by changing the locations of the distractors in addition to adding more fake answers. [273] also modify the approach by further shuffling the distractor and find that BERT's performance drops by 50%. [169] propose three new methods to generate QA adversaries which result in up to 45% performance drop, while [253] use context shuffling and find that the F1 scores of models decrease slightly. [33] generate fluent and grammatical adversarial contexts which lower model confidence on the gold answer or direct the model towards an incorrect answer, and [274] use character swapping and paraphrasing and show that state-of-the-art models are vulnerable. [2] use random, structural, and irrelevant noise, and find that a sufficient amount of noise can reduce the performance by 70%. [153] focus on typos, such as capitalization or common misspellings, while [250] use adverb modifications and find that models struggle with most of them. Lastly, [268] add an irrelevant sentence to the context which results in a dramatic decrease model performance.

The discussed work highlight that: 1) models can be easily dissuade by many types of distractors, even those that are nonsensical; and 2) the type and complexity of the distractor matter

and can result in either minimal or substantial performance drop.

Proposed Approaches A popular approach to improve models' robustness to distractors is to train with augmented noisy data [239, 301, 169, 19, 183, 78, 192, 33, 274, 132, 150]. But some suggest that this has limited benefits [109, 301, 274]. Another possibility is to train models to edit distractor information, as done in [18], or to prompt systems to ignore irrelevant information [268].

4.2.3.4 Conflicting Knowledge

Next, we focus on Rows 3 and 4 in Table 4.7: contexts with information that is conflicting with the original context. The question is typically: how susceptible are systems to contexts that conflict with their parametric knowledge? While the alternate context conflicts with models' parametric knowledge in the known knowledge split, this is not necessarily the case for the unknown knowledge split, as the alternate context may already be contained within the model's parametric knowledge.

Overview The most popular approach to evaluate systems on conflicting knowledge is entity substitution. [163] replace the original answer entity with either a similar type one, an alias, an entity from the same corpus, or an entity based on popularity. This allows them to discover many aspects that affect models' over-reliance on their parametric knowledge, such as their size and domain. [349] use a similar approach and focus on improving the robustness of systems to conflicting knowledge using prompts. [34] modify the approach and use multiple contexts, and find that the performance of the IR system has a large effect on whether a model will use parametric knowledge. [199] use the same approach but focus on disentangling systems' parametric and contextual knowledge, while [99] find that models are very brittle to conflicting information in both in-context few-shot learning and fine-tuning settings. [67] find that models are approximately $3-4$ F1 points worse with conflicting entities, but also mention that such substitution can also affect the context's grammar. [321] propose to use entities of different implications, while, [84] find that models' performance can be reduced by up to 25% with conflicting entities.

The second most popular approach is to use negations. [90] automatically create contexts that are pragmatical specifically for each Transformer model, and find that most models are sensitive

to negation. [253] find that models continuously predict the original answer with negations, and [127] find that models often think that negative facts are true. Other methods also exist, such as using Mechanical Turkers [210] or graduate students [293], which result in a significant performance change.

Some also use a masked language model to create conflicting knowledge [210, 152], where the former find that models are vulnerable to contradicting contexts, the latter mention that such an approach results in fluent and semantically preserving context. [212] use GPT-3.5 to generate conflicting contexts which result in a significant decline in system performance, while [150] use T5 [228] and find that model's robustness does not scale with a model's size increase. [346] randomly replace objects and find that models fail on conflicting multi-hop questions, while [222] train a neural perturbation model to modify demographic terms. Lastly, [84] change the order the events or dates and find that model performance is greatly reduced.

The discussed work highlight that: 1) systems over-rely on their parametric knowledge, which often result in knowledge conflicts; 2) the type of conflicting information matters, but not necessarily for the right reasons. For example, [67] find that entity substitution can affect the context's grammar, which can in general result in a decrease in performance.

Proposed Approaches As our knowledge is changing, [351] propose the task of modifying factual knowledge specifically in Transformer models [295], while [52] use a hyper-network to predict the weight update of systems. [189] use a collection of auxiliary networks that update a pretrained model's behavior, and [179] identify factual-relevant neuron and update their weights. [99, 212, 210] propose a misinformation detector, but the latter mention that the benefits are limited with insufficient training data. [316] mention that improving the coherence of the context can improve the receptiveness of LMs to it, while [163] suggested to use a perfect retriever or to augment the training data with conflicting knowledge. [132, 222, 150, 293, 72, 34] also suggest to train with data augmentation, but the latter mention that it does not easily generalize to other methods of creating conflicting knowledge. [272, 349, 212] mention that carefully designed prompting strategies can improve the performance, while [199] suggest that models should generate two answers – a

parametric one and a contextual one. [346] propose to store all edited facts externally, while [321] propose entity-based masking. Lastly, [353] propose to use a natural language inference component to detect contradiction.

4.2.3.5 Models' Parametric Knowledge

Next, we focus on Row 5 in Table 4.7: an empty context with no distractor. This is the standard setting for evaluating model-internal knowledge or for determining whether models are "knowledge bases" [219]. The question is: which facts are known or unknown to the model?

Overview Recently, the size of LMs, which are the basis of recent state-of-the-art QA models, has been increasing dramatically [295, 224, 225, 42, 304]. This in turn allows them to remember a massive amount of factual knowledge [219, 85, 240, 126, 53, 281, 114, 269].

There are several ways to evaluate a model's parametric knowledge. For example, [345, 269, 126, 281, 219, 113, 58, 205] use "fill in-the-blank" cloze statements, [150, 316, 240] use a closed-book setting, and [47] expand a knowledge graph around a seed entity by prompting the system.

The success of such models to recall factual information allows them to be useful in tasks that require knowledge, without supplying them with actual context [128], and even becoming competitive with other state-of-the-art fine-tuned models [30]. However, training systems to memorize facts may also have adverse results. Systems have been shown to often ignore the context and focus on their parametric knowledge [163, 129, 195]. This results in hallucinations [163], and poor performance when the knowledge is different than the training data [150, 199, 163].

The discussed work highlight that: 1) there is no one correct approach to evaluate systems' knowledge; 2) developing systems with more knowledge is not necessarily better. For example, in domains where knowledge is often changing, it might be more important for systems to be more flexible to different contexts than knowledgeable, such as in medicine, where new treatments often arise. **Proposed Approaches** While many evaluate parametric knowledge, not many *directly* focus on increasing it. However, existing experiments show that bigger models or different architectures can help [219, 240]. Furthermore, better knowledge can also be learned via multimodal

training [10].

4.2.3.6 Irrelevant Knowledge

Next, we focus on Row 6 in Table 4.7: irrelevant context. The question is: how often does a system changes its answers when given irrelevant context?

Overview What we define as irrelevant context exists in many datasets, such as the Natural Questions [138], SQuAD 2.0 [232], QuAC [39], CoQA [238], and MS MARCO [17], where the answer to the question is not supported by the context. Other work also evaluate such irrelevant context formulation, such as [150], which define irrelevant contexts as those that do not entail the answer. They find that models are strongly interfered by irrelevant contexts, especially those that share a similar general topic as the question. Additionally, [199] find that random irrelevant context is more challenging to models in some settings.

As the field is moving towards using LLMs which contain large amount of knowledge, these type of questions become not truly "unanswerable" with irrelevant context, or even without any context, which further reinforce the need to split existing evaluations into known and unknown knowledge. This is in comparison to subjective, philosophical, or imagination questions such as proposed in [332], which are truly "unanswerable" by any system, regardless of their knowledge.

Proposed Approaches While we discuss many approaches to improve model robustness on contexts with added distrators in Section 4.2.3.3, not many evaluate models using irrelevant context of our setting – based on known and unknown knowledge. [150, 199] propose training with data augmentation, while the latter further train the model to disentangle its parametric and contextual knowledge by generating two answers.

4.2.4 Defining Desiderata

4.2.4.1 Problem Formulation

Given a dataset composed of questions $q_1, q_2, ..., q_n \in Q$, their corresponding answers $a_1, a_2, ..., a_n \in A$ and contexts $c_1, c_2, ..., c_n \in C$, we evaluate how well a model uses the context or its modifications,

which will be presented in the following sections, on the given questions.

We note that two types of context-based QA datasets exist: 1) the questions are about a general knowledge concept and the contexts supplement the knowledge, for example, *"Who is the current president?"* Relevant datasets are, e.g., WikiQA [323], SQuAD 1.0 [234], and OpenBookQA [184]; and 2) the questions are specifically about the contexts, for example, *"What did the narrator mean by [...]"*. Datasets include, e.g., Race [139], QuAIL [241], and CosmosQA [104]. **We only use the first type**, as those questions can be used to measure models' parametric knowledge by omitting the context, while the latter cannot, as without context the models cannot answer the question.[4]

4.2.4.2 Creating Conflicting Context

We follow [210, 152]'s approach of using a masked language model. More formally, we mask the answer string a_i from the context q_i when it exists verbatim.[5] We then use DistilBERT [247] to predict the masked answer, and replace the masked token with it. For each masked answer we generate 10 different answers, and remove any that are similar (i.e., exact string match) to the original answer. This results in up to 10 conflicting contexts for each question. In the free-form setting, we then replace the original answer a_i with the new predicted answer. For the MC setting, we leave the original answer as one of the MC options and replace one wrong answer with the new answer. The ideal behavior of systems for such context can be seen in Table 4.7 in Rows 3 and 4.

4.2.4.3 Creating Irrelevant Context

What we define as irrelevant context exists in many datasets, such as those described in Section 4.2.3.6. These type of questions have been termed "unanswerable" questions. However, in our formulation, if the context is irrelevant or does not exist, models may still have the parametric knowledge to answer the corresponding question (e.g., from pretraining), which makes these questions not truly "unanswerable." **We avoid using such datasets as the correct answer is not**

[4] If they do, it is by mere chance.
[5] The answer string is sometimes paraphrased in the context. We discard such questions.

provided (e.g., SQuAD 2.0 has empty strings as labels for its irrelevant contexts), which prevents us from determining if the model has the parametric knowledge to answer.

To create irrelevant context we opt for a method that can be applied to most existing context-based QA dataset and follow [199]'s approach of selecting random contexts. More formally, for each question q_i, we replace the corresponding context c_i with a random context $c_j \in C$, where $c_i \neq c_j$. We repeat this 5 times which results in 5 irrelevant contexts for each question.[6] The ideal behavior of systems for such context can be seen in Table 4.7 in Row 6.

4.2.4.4 Context with Distractor

To add a distractor to contexts we use the ADDANY approach [109], but modify it to be applicable to free-form and MC settings. In particular, instead of modifying w_i to be the x that minimizes the expected value of the F1 score, we update it to be the one that maximizes the perplexity of the answer with respect to the input string in the free-form setting, and the one which minimizes the probability of the correct answer for MC.

4.2.5 Experiments

In our experiments, each context c_i and question q_i are input into the model within the following string: "question: q_i. context: c_i." In the free-form setting and for MCQA, we use exact match (EM) and, respectively, accuracy to measure model performance. That being said, our approach is by design extremely easily adaptable to different choices of metrics, such as LLM-based ones [120], which could have a higher correlation with humans for QA tasks than EM.

4.2.5.1 Datasets

We experiment on 5 datasets: 1) SQuAD 1.0 [234], ,[7] 2) AdversarialQA [19], which we both use for free-form QA, where the answer span is to be generated, as well as 3) Natural Questions [138]. Additionally, we also use 4) SciQ [308] and 5) MedMCQA [209], which are MCQA datasets.

[6] While there is a small chance that the random context contain some relevant information, it is unlikely.
[7] The reason we use SQuAD 1.0 and not the later version is discussed in Section 4.2.4.3.

4.2.5.2 Models

We evaluate 5 LLM-based QA models in the free-form setting: GPT 3.5, GPT 4 [207], BART [145] base, T5 [228] small, and six LLM-based QA models in the MCQA setting: BERT [56] base, BigBird [333] base, Longformer [20] base, RoBERTa [160] base, ALBERT [140] base, and DistilBERT [247] base. We finetune each pretrained model on the training set for 20 epochs, use early stopping on the validation with patience of 3, and evaluate them on the test set. As the test sets for SQuAD 1.0 and Natural Questions are not publically available, we split the validation set into 2 for all models, and use one half as the test set. Lastly, on the Natural Questions dataset we evaluate 3 published models (without further training) from [240] to analyze how they score on our desiderata. These models are 1) T5-Small,[8] 2) T5-Large-1.0,[9] and 3) T5-Large-0.9.[10] .

4.2.5.3 Results and Analysis

Our toolkit takes most context-based datasets, as described in Section 4.2.4.1, and automatically prepares and evaluates all desiderata aspects *at once*. We use it to evaluate each of the models described in Section 4.2.5.2 in all of the settings shown in Table 4.7. In comparison to previous work, we split desiderata aspects by finding the context that is known and unknown to individual models, as the ideal behavior of models' depends on if the knowledge contained in the context is known or unknown to the model. Our results can be seen in Tables 4.8 and 4.9.

Amount of Knowledge We calculate the amount of knowledge models possess using the closed-book setting and accuracy, as described in Section 4.2.4.1. On the SciQ and MedMCQA datasets, models possess sufficient knowledge to accurately respond to approximately half and one-third of all queries, respectively, without additional context. Interestingly, ALBERT performs the poorest on both datasets, achieving an accuracy rate of 45.3% on SciQ and 22.7% on MedMCQA. In contrast, BigBird and Longformer score the highest on SciQ and MedMCQA, with accuracies of

[8] https://huggingface.co/google/t5-small-ssm-nq

[9] T5 large that is fine-tuned on 100% of the train splits of Natural Questions. https://huggingface.co/google/t5-large-ssm-nq

[10] T5 large that is fine-tuned on 90% of the train splits of Natural Questions. https://huggingface.co/google/t5-large-ssm-nqo

56.4% and 32.3%, respectively. This aligns with previous discussed work in Section 4.2.3.5, which suggest that such models contain abundant factual information and have the potential to be used as open-domain QA systems.

In comparison, the free-form models could not answer even 9% of the questions successfully without context (GPT-4 scores 8.7% on SQuAD).[11] , The significant difference in performance between the MC and the free-form models may partially be due to the fact that the MC setting is much easier, where a model that randomly predicts an answer gets on average 25% of the questions correctly.[12]

The Standard Evaluation Almost all models (except for ALBERT on MedMCQA and T5-small on SQuAD) score higher on the known vs. the unknown knowledge split. For example, 99.1% vs 96.6% for BigBird on SciQ and 58.4% vs 4.8% for GPT-4 on AdversarialQA. This suggests that models find context that reinforce their knowledge beneficial, which emphasize that future work should evaluate systems from knowledge perspective.

Distractor Similar to previous work (cf. Sec 4.2.3.3), we find a significant reduction in performance across all MC models (e.g., on SciQ, DistilBERT's performance drops from 97.4% to 4.0% on known knowledge). Furthermore, the difference between known and unknown knowledge is visible, where across almost all models (except for DistilBERT on SciQ, and Longformer and ALBERT on MedMCQA) noise affect unknown knowledge more. While there is also a clear reduction in performance for free-form models, the reduction is not as large. For example, T5 small drops from 72.6% in the unknown knowledge split to 68.9%.

Conflicting Knowledge We also find a substantial performance drop across all models when conflicting knowledge is introduced. For example, 33.2% for RoBERTa in the known knowledge split on SciQ, and 50.0% for GPT 3.5 in the known knowledge split on AdversarialQA. We also find again, a difference in behavior across almost all MC models between known and unknown knowledge: the performance drop is lower in the unknown split, which we believe occurs as, for the

[11] Due to the small number of correct instances, we cannot draw any strong conclusions regarding such systems in the known vs. unknown knowledge splits.

[12] We also try non-finetuned versions of the free-form models, but the results are comparable.

known knowledge split, this type of substitution conflicts with the model's parametric knowledge, while this might not be the case for the unknown split as discussed in Sec 4.2.3.4.

Irrelevant Context We find that all models are more consistent with their answers for known knowledge when irrelevant contexts are added. For example, T5-base generates similar answers to 65.1% of the questions for known knowledge and only 21.3% to questions for unknown on SQuAD, while Longformer generates similar answers to 53.5% vs 50.2% for known and unknown knowledge on MedMC, respectively. This might suggest that systems are more confident about known information and hence are less likely to change answers.

Distractor + Conflicting Knowledge Combined Looking at the *combination* of distractors with conflicting contexts, we find that the performance drop is generally lower in the unknown split for most models. We can also see that the combination of conflicting contexts and added distractor can result in accuracy drop of close to 96%, such as in DistilBERT in known knowledge on SciQ.

Distractor + Conflicting Knowledge – Separate Looking at the models' performances in the conflicting knowledge and distractor addition settings *separately*, we can further see that systems that are more susceptible to noise are often more likely to correctly answer according to a context that conflicts with their parametric knowledge. For example, within the MC systems, DistilBERT has the largest performance decrease for added distracor, but also performs nearly the best on conflicting knowledge on SciQ. Similar trends can be seen between ALBERT and RoBERTa, Longformer and RoBERTa, BERT and BigBird, and others. A potential reason might be that the susceptibility of systems to noise occurs as they are more attentive to everything in the context, which is beneficial for conflicting knowledge.

Distractor + Consistency Looking at models' performances for the distractor and irrelevant context settings, we find that systems that are less susceptible to distractors are not necessarily more consistent with their answers when provided irrelevant context. BigBird is the more susceptible to distractors than Longformer on SciQ, and less consistent than it for unknown data, where opposite trends occur between BigBird and Longformer.

MC vs. Free-form For added distractors, we find that MC models are more susceptible than the free-form ones, and have a larger performance drop. This may be due to the fact that such models are less susceptible to noise, or that the optimization method we use to find noisier sentences in the free-form is not as strong as the one we apply in the MC setting (Section 4.2.4.4). For conflicting knowledge, the reduction in performance between the MC models and the free-form ones is also visible and somewhat comparable. For example, GPT-4 score is reduced by 53.9% on the known knowledge split when conflicting knowledge is added on AdvarsarialQA, in comparison to BigBird's performance on MedMCQA decreases by 36.0%.

Model Size We also test similar types of models in two sizes: T5-small and T5-base, and GPT-3.5 and GPT-4. We find that the larger variant 1) has a larger amount of known knowledge. For example, the T5 models score 0.9% vs 0.3% on SQuAD and 4.2% vs 2.9% on AdvarsarialQA, where the GPT models score 8.7% vs 0.3% on SQuAD and 5.9% vs 0.2% on AdvarsarialQA; 2) is more robust to distractors. For example, T5-base decreases by 6.0% on known knowledge on SQuAD, where the smaller version decreases by 10.0%; 3) is not necessarily more robust to conflicting knowledge on known knowledge. For example, GPT-4's performance drop is larger than GPT-3.5 on SQuAD, but T5-base's drop is lower than T5-small on the same dataset; 4) is not necessarily more consistent with its answers. For example, T5-small is more consistent for unknown knowledge on SQuAD and AdvarsarialQA, but less consistent for known knowledge. Oppositely, GPT-4 is more consistent for unknown knowledge on AdvarsarialQA, but less consistent on SQuAD.

Setting/ Emphasis	*Question: <q> Context: <c>*	*Context: <c> Question: <q>*
Question	Question: **where is the world's largest ice sheet located today.** Context: The Antarctic ice sheet is the largest single mass of ice on Earth [...]	Context: The Antarctic ice sheet is the largest single mass of ice on Earth [...]. Question: **where is the world's largest ice sheet located today.**
Context	Question: where is the world's largest ice sheet located today. Context: **The Antarctic ice sheet is the largest single mass of ice on Earth [...]**	Context: **The Antarctic ice sheet is the largest single mass of ice on Earth [...].** Question: where is the world's largest ice sheet located today.
Question+ Context	Question: **where is the world's largest ice sheet located today.** Context: **The Antarctic ice sheet is the largest single mass of ice on Earth [...]**	Context: **The Antarctic ice sheet is the largest single mass of ice on Earth [...].** Question: **where is the world's largest ice sheet located today.**

Figure 4.1: Example from the Natural Questions dataset in which we show the different settings we experiment with: question or context first in the input prompt, and the different substring emphasis (in bold). $< q >$=question string; $< c >$=context string.

Model	**Emph.**		**Natural Questions**								**SQuAD**								**AdversarialQA**							
			Question First				Context First				Question First				Context First				Question First				Context First			
			No Emph.	Emphasis			No Emph.	Emphasis			No Emph.	Emphasis			No Emph.	Emphasis			No Emph.	Emphasis			No Emph.	Emphasis		
				Q	C	Q + C		Q	C	Q + C		Q	C	Q + C		Q	C	Q + C		Q	C	Q + C		Q	C	Q + C
Llama-2	B		46.3				58.1				60.4				72.9				42.6				49.4			
	AS			54.8	53.0	-		57.8	59.3	-		66.3	62.0	-		74.5	72.9	-		43.3	43.0	-		54.4	53.3	-
	MP	⋆		51.4	31.6	53.1		58.3	56.4	58.8		56.8	61.7	67.9		69.1	76.4	79.7		40.5	43.2	46.7		51.1	54.2	57.5
		"		48.7	54.2	54.2		56.4	58.2	59.9		61.4	71.9	72.3		72.5	76.3	78.6		42.0	48.3	48.9		50.2	56.8	56.0
		<mark>		51.7	54.1	55.1		60.0	55.5	60.5		53.3	71.5	71.8		75.4	71.3	80.4		39.0	47.3	49.3		50.7	52.4	**57.7**
		<emphasize>		47.6	54.4	53.9		**61.3**	55.5	60.2		53.8	72.2	68.0		78.1	70.4	**81.5**		37.8	49.3	46.5		51.4	50.4	56.2
GPT-J	B		33.3				64.5				45.5				61.0				47.2				46.2			
	AS			66.3	66.3	-		61.1	53.0	-		51.0	44.6	-		55.8	54.1	-		45.0	37.8	-		41.6	41.7	-
	MP	⋆		33.4	26.9	49.7		60.5	65.1	64.9		38.0	52.5	41.7		51.1	64.0	50.5		38.2	52.0	40.8		40.2	50.0	38.2
		"		39.0	63.0	62.3		66.3	65.9	66.7		34.0	56.2	49.5		61.7	61.0	66.4		35.8	53.4	50.2		48.7	49.7	52.5
		<mark>		34.3	61.6	52.9		61.5	67.8	64.4		40.5	64.2	55.9		66.8	68.5	**72.3**		41.8	64.1	52.2		57.4	55.0	60.2
		<emphasize>		38.3	**69.0**	64.2		62.7	63.6	62.9		37.1	64.7	55.9		65.0	68.1	69.5		38.4	**64.8**	57.7		57.0	52.0	59.5
GPT-2 Large	B		34.0				44.5				27.1				42.3				27.7				26.9			
	AS			**63.2**	54.8	-		54.7	45.1	-		54.5	45.2	-		46.0	43.7	-		**58.4**	44.9	-		32.8	33.6	-
	MP	⋆		22.1	44.9	30.5		43.4	42.2	41.2		23.7	30.1	39.2		39.9	43.8	44.0		22.6	30.0	38.0		25.2	27.8	27.7
		"		29.7	41.2	41.8		40.0	40.9	44.0		27.3	31.3	36.8		42.5	47.3	49.1		27.9	30.4	32.2		27.7	32.3	30.0
		<mark>		35.4	46.1	34.1		35.6	45.8	25.1		25.6	**56.5**	51.1		36.1	48.4	42.0		26.0	57.5	50.5		22.5	31.6	27.4
		<emphasize>		34.8	46.7	45.4		38.2	45.8	30.3		26.3	52.2	55.1		40.8	47.7	44.3		25.4	51.1	55.6		25.4	30.6	27.0
GPT-2 XL	B		28.0				51.2				20.5				50.1				24.8				31.8			
	AS			34.0	39.9	-		**55.9**	45.7	-		35.5	25.6	-		52.5	52.4	-		33.9	34.9	-		36.3	34.6	-
	MP	⋆		28.9	31.0	41.7		48.7	48.1	49.3		21.1	25.7	32.1		49.5	51.2	50.2		23.2	27.2	28.1		31.8	34.0	33.8
		"		30.2	35.8	43.7		50.0	46.0	46.1		23.5	29.9	37.5		49.8	51.8	51.9		25.6	28.2	30.7		32.2	33.6	33.6
		<mark>		30.1	43.3	51.0		49.8	49.5	47.0		17.4	38.3	36.2		47.3	**53.4**	49.9		19.0	**38.1**	34.8		29.8	34.8	31.6
		<emphasize>		28.4	42.3	42.9		48.2	50.4	46.1		18.2	32.3	37.9		48.7	**53.4**	50.4		20.8	32.1	34.2		30.0	35.2	32.4

Table 4.1: Question vs. Context Table: B=Baseline (no emphasis); AS=Attention steering; MP=Marked prompting; C=Context; Q=Question; <q>=question string; <c>=context string; The highest score for each model is in bold, the second highest on the other prompt structure is underlined. The AS method requires a substring within the input string to be emphasized, and hence, it is undefined for the Q+C setting, as in that setting the substring will be the entire input string.

Model	NQ		SQuAD		AdversarialQA	
	Question First	Context First	Question First	Context First	Question First	Context First
Llama	**15.08**	15.53	11.49	**10.58**	12.89	**11.96**
GPT-J	20.16	**18.61**	13.13	**13.07**	14.52	**14.36**
GPT-2 Large	36.26	**32.22**	**20.86**	21.24	**22.88**	23.29
GPT-2 XL	30.44	**28.47**	19.02	**18.89**	20.99	**20.70**

Table 4.2: Model's average perplexity on each dataset, for each prompt structure, in the zero shot (no emphasis) setting. Lower is better. NQ=Natural Questions.

Model	Emphasis Method	Question Emphasis		Context Emphasis	
		Accuracy	Question String Avg. Attention Score	Accuracy	Context String Avg. Attention Score
GPT 2 Large	*	22.1	0.0078	44.9	0.0041
	"	29.7	0.0078	41.2	0.0094
	mark	35.4	0.0074	46.1	0.0088
	emphasis	34.8	0.0070	46.7	0.0084
GPT 2 XL	*	28.9	0.0076	31.0	0.0039
	"	30.2	0.0075	35.8	0.0095
	mark	30.1	0.0071	43.3	0.0089
	emphasis	28.4	0.0067	42.3	0.0085

Table 4.3: Attention scores analysis across different models' layers and heads for different emphasis methods.

Model	Natural Questions					SQuAD					AdversarialQA				
	Kn. Amount	Known No Emph.	Known Emph.	Unknown No Emph.	Unknown Emph.	Kn. Amount	Known No Emph.	Known Emph.	Unknown No Emph.	Unknown Emph.	Kn. Amount	Known No Emph.	Known Emph.	Unknown No Emph.	Unknown Emph.
Llama 2	20.0	93.4	93.6	**46.4**	**49.9**	18.1	88.6	91.9	**70.0**	**79.7**	20.5	77.9	71.7	**42.7**	**51.9**
GPT J	4.3	90.2	89.5	**63.2**	**65.5**	9.2	83.5	86.5	**58.7**	**71.2**	14.2	71.3	73.7	**42.0**	**58.0**
GPT 2 Large	1.7	**78.8**	**86.4**	43.7	43.1	4.6	79.6	84.1	**40.5**	**47.6**	11.4	64.9	61.9	**22.0**	**25.9**
GPT 2 XL	2.2	88.5	85.2	**50.2**	**48.4**	6.0	78.5	79.1	**48.2**	**50.3**	11.9	**67.5**	**69.8**	26.9	28.9

Table 4.4: Known vs. Unknown Table: **Marked Prompting**. We find that the best emphasizing method is marked prompting, and in particular, concatenating the string "<emphasize>" before and after the context and question strings. We use the closed-book setting to evaluate models' parametric Kn., and compare the ZS baseline (no Emph.) to the best marked prompting approach. In bold, the largest improvement for each model on each dataset. Kn. Amount is measured using accuracy, as the average number of questions models can successfully answer correctly without context (cf. Section **??**).

Model / Dataset	Natural Questions				SQuAD				AdversarialQA			
	Known No Emph.	Known Steering	Unknown No Emph.	Unknown Steering	Known No Emph.	Known Steering	Unknown No Emph.	Unknown Steering	Known No Emph.	Known Steering	Unknown No Emph.	Unknown Steering
Llama-2	**69.1**	**81.0**	30.8	37.0	80.6	85.8	**56.7**	**62.0**	69.8	67.8	**38.0**	**38.5**
GPT-J	56.4	76.8	**27.9**	**59.9**	53.4	66.5	**44.7**	**62.6**	**63.8**	**63.8**	44.4	41.9
GPT-2 Large	52.1	71.4	**29.4**	**54.6**	47.1	68.9	**26.1**	**53.7**	42.1	59.6	**25.8**	**58.3**
GPT-2 XL	51.4	49.1	**24.2**	**33.5**	39.1	52.4	**19.3**	**34.4**	**40.7**	**50.8**	20.9	29.6

Table 4.5:]

Known vs. Unknown Table: **Attention Steering**. While attention steering does not overall perform as well as marked prompting, we also evaluate models' parametric knowledge (known vs. unknown) using the closed-book setting, and compare the ZS No Emph. (no Emph.) to the attention steering approach where the question is presented first in the prompt and is being emphasized – as that is the best setting we find for attention steering. In bold, the largest improvement for each model on each dataset.

Model	Emphasis Method	Natural Qustions					
		Question First			Context First		
		No Emphasis	Q	C	No Emphasis	Q	C
Falcon-7B		17.0			40.2		
	*		10.2	12.8		25.0	38.6
	"		17.0	36.8		38.0	**42.4**
	mark		11.0	34.0		34.4	41.4
	emphasis		9.8	30.8		36.0	40.2
Falcon-7B Instruct		24.4			39.8		
	*		24.6	17.4		20.8	40.6
	"		29.0	34.2		16.6	36.2
	mark		25.0	**47.2**		16.0	39.8
	emphasis		16.2	42.6		12.6	38.6
MPT-7B		17.0			43.5		
	*		20.5	18.5		49.0	**53.5**
	"		16.0	42.0		36.0	37.5
	mark		34.5	49.5		29.0	46.0
	emphasis		17.5	37.5		38.0	52.0
MPT-7B Instruct		25.0			13.0		
	*		26.7	20.2		32.0	14.0
	"		15.7	29.2		8.25	12.5
	mark		15.0	26.2		15.0	13.0
	emphasis		20.5	**40.7**		20.5	13.0
Llama-13B		28.4			58.6		
	*		27.4	27.2		41.2	55.8
	"		30.4	55.0		52.0	57.0
	mark		23.4	36.8		41.6	60.0
	emphasis		26.4	53.4		49.4	**60.8**

Table 4.6: Analysis of newer models, two of which are instruction-tuned, where all models are evaluated using their maximum context length (up to 4k).

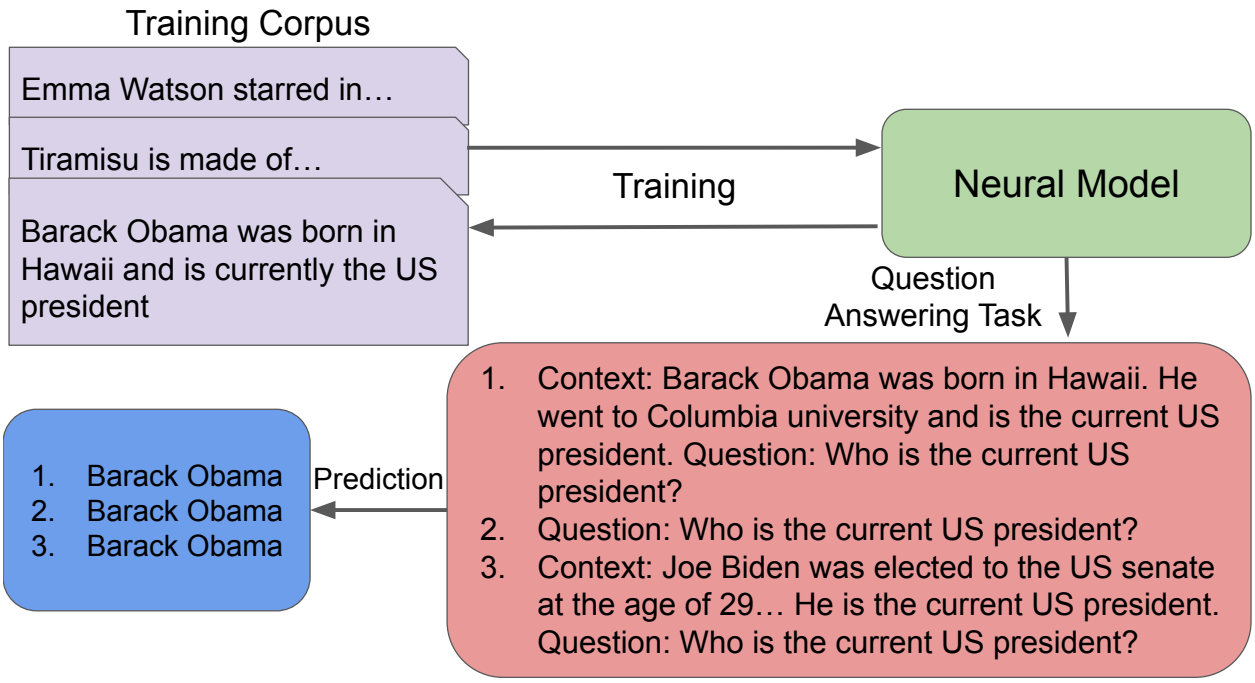


Figure 4.2: An example where the model was trained to learn the knowledge "Barack Obama is the current US president". In the first and second tasks the model answered the questions correctly. However, in the third task where the model is given context with conflicting information it fails to answer the question correctly.

	Context	Distractor	Known Knowledge	Unknown Knowledge
1	Original		T	T
2	Original	✓	T	T
3	Alternative		A	A
4	Alternative	✓	A	A
5	None		T	B
6	Irrelevant		T	B

Table 4.7: Desiderata table: what should an optimal model do for different types of contexts? T = *true answer*; A = *conflicting answer*; B = *wrong answer/unanswerable*; The distractor (cf. Sec. 4.2.3.3) is a string of words that is concatenated to the context (cf. Sec. 4.2.4.4); Alternative context (cf. Sec. 4.2.3.4) is a slight modification of the original context, where we replace the answer string with an alternative one (cf. Sec. 4.2.4.2); Irrelevant context (cf. Sec. 4.2.3.6) is a random context (cf. Sec. 4.2.4.3).

Dataset	Model	K. Am.	St. KK	St. UK	St. Avg	Dist. KK	Dist. UK	Conf. KK	Conf. UK	Conf. Dist. KK	Conf. Dist. UK	Irr. KK	Irr. UK
SciQ	BERT	52.7	97.4	95.2	96.3	56.5	46.3	63.1	69.7	29.8	**34.9**	**82.8**	**73.8**
	BigBird	**56.4**	99.1	96.6	97.9	36.7	23.6	**75.2**	**78.6**	11.0	13.1	78.9	60.4
	Longformer	55.4	**99.5**	**98.4**	**99.0**	**71.4**	**61.5**	66.3	71.4	**31.2**	34.2	81.4	68.4
	RoBERTa	51.9	**99.5**	96.7	98.1	20.0	9.0	73.4	77.9	17.0	7.3	76.6	65.1
	ALBERT	45.3	99.2	97.1	98.1	55.0	43.7	69.4	74.6	20.0	25.9	80.5	71.4
	DistilBERT	49.6	97.4	94.8	96.1	4.0	4.0	73.0	76.5	1.0	1.0	67.9	61.7
MedMC	BERT	31.1	84.1	**81.6**	82.9	75.7	**64.3**	56.5	**61.8**	**73.5**	61.3	66.7	61.6
	BigBird	27.3	83.9	74.5	79.2	65.5	51.9	47.9	55.4	21.3	32.5	58.8	53.0
	Longformer	**32.3**	84.9	78.4	81.7	61.7	61.1	53.2	55.0	58.7	**70.6**	53.5	50.2
	RoBERTa	28.7	**88.3**	81.2	**84.7**	**76.6**	60.5	62.1	60.6	73.0	64.6	59.4	51.7
	ALBERT	22.7	76.6	77.9	77.3	41.6	62.1	39.8	42.3	38.4	58.1	38.8	34.8
	DistilBERT	28.8	84.1	76.6	80.3	63.3	53.9	**62.5**	60.0	40.5	46.3	**66.9**	**62.3**

Table 4.8: Results table: MCQA models. K. Am=Knowledge amount; St=Standard; KK=known knowledge; UK=unknown knowledge; Dist=distractor; Conf=conflicting; Irr=Irrelevant. Each model's parametric knowledge results in different known and unknown knowledge splits which we evaluate using accuracy. In bold, highest accuracy on each of the desiderata components for each dataset.

Dataset	Model	K. Am.	St. KK	St. UK	St. Avg	Dist. KK	Dist. UK	Conf. KK	Conf. UK	Conf. Dist. KK	Conf. Dist. UK	Irr. KK	Irr. UK
SQuAD	T5-Small	0.3	70.0	72.6	72.6	60.0	68.9	53.1	63.5	53.1	55.4	45.9	**25.8**
	T5-Base	0.9	**82.0**	**78.4**	**78.4**	**76.0**	**75.4**	**75.6**	**64.7**	**70.7**	**61.3**	**65.1**	21.3
	BART	0.9	68.7	65.4	65.4	60.4	59.9	55.0	50.2	51.3	43.0	48.3	24.0
	GPT-3.5	0.3	50.0	0.3	0.4	-	-	33.3	0.1	-	-	2.7	2.6
	GPT-4	**8.7**	45.3	10.4	13.4	-	-	12.1	6.5	-	-	32.3	0.6
Adv. QA	T5-Small	2.9	63.6	20.1	21.4	59.0	19.3	6.0	16.5	4.3	14.6	**69.6**	**24.9**
	T5-Base	4.2	65.0	**27.2**	**28.8**	60.3	**37.7**	**11.8**	**19.2**	**10.5**	**20.5**	57.1	5.4
	BART	4.1	**87.0**	20.2	23.0	**77.4**	16.7	9.2	11.8	6.7	7.6	60.2	13.6
	GPT-3.5	0.2	50.0	2.4	**2.5**	**-**	-	0.0	0.5	-	-	50.0	0.9
	GPT-4	**5.9**	58.4	4.8	8.0	-	-	4.5	1.5	-	-	41.5	11.9

Table 4.9: Results table: free-form models. K. Am=Knowledge amount; St=Standard; KK=known knowledge; UK=unknown knowledge; Dist=distractor; Conf=conflicting; Irr=Irrelevant. Each model's parametric knowledge results in different known and unknown knowledge splits which we evaluate using accuracy. In bold, highest accuracy on each of the desiderata components for each dataset. The distractor setting is not done for the GPT models as it requires model accesss.

Chapter 5

Model Improvement

Building on the desiderata, I expand my dissertation work to the development of improved models, aiming to address the issues identified in my previous research and enhance the performance, reliability, and interpretability of QA systems.

5.1 Who Are All The Stochastic Parrots Imitating? They Should Tell Us!

The work described in this section has been published in AACL 2023 [257].

5.1.0.1 Introduction

Transformers [297] and related models have been improving rapidly, with applications in a surprisingly large number of domains, such as natural language generation [337], machine translation [300], question answering [1], and code generation [283], based on the ability to generate sensible outputs to prompts over a nearly limitless input domain.

Despite impressive performance on a wide array of benchmark tasks, these models are known to produce "AI-splaining," confident sounding but incorrect statements: "To the extent that a use case places importance on the truth of the outputs provided, it is not a good fit for GPT-3" [49]; see also [45] and [174].

This problem has proven to be especially true for models trained on low-resource languages [91], where data may not only be scarce [168], but also not well curated with respect to correctness or quality, in comparison to higher-resource languages [96]. Furthermore, model hallucination in

Figure 5.1: An actual conversation with ChatGPT in Hebrew on the effects of not drinking enough water. ChatGPT is unable to point the user to its sources and instead falls back to a general answer ("I am ChatGPT, an OpenAI model based on the GPT-3.5 deep learning model. I am powered by OpenAI's learning set, which has been raised with the help of machine learning techniques on Internet culture, including websites, books, articles, quotes, and more"). We argue that ChatGPT and similar models should be able to direct the user to the sources of their information, which will have multiple benefits, such as quick verifiability of model statements.

such settings can result in toxic patterns that can be found in the training data [91].

In accordance with the large LMs and low-resource languages theme track, we argue that while the performance and factuality of LMs has been improving, both in high-resource and low-resource settings, in their existing state, LMs will realistically never be fully trustworthy. Thus, in settings in which factuality is required, such as medicine, they are dangerous and unemployable. This is further noted in [181], who state that users cannot trust any claim a model makes without fact-checking.

Our proposal to address these concerns suggests both technical development and a simple regulatory framework: as we often ask students, journalists and scholars, **we should ask our models to name their sources and provide evidence for their assertions.** Currently, even popular LMs often fail at this, as seen on the ChatGPT example in Figure 5.2. In the case of generative models, either the model itself or a post-hoc procedure could – and, under certain circumstances, **should be required to** – be designed to produce evidentiary justification for its output.

NLP tasks would benefit from such citation models, discuss the benefits they would bring, and present a roadmap to develop such models. Our goal is to motivate the field to start thinking

about what is necessary to make current models truly useful in all sorts of – potentially critical – scenarios.

5.1.1 Background

Factuality and the Lack Thereof LMs store factual knowledge [59, 255, 54, 68, 112] and previous work have shown that LMs can act as knowledge bases [220, 282]. However, there is no guarantee that the retrieved knowledge is indeed factual, and unfortunately, often it is not. This can be seen in many areas, such as question answering [319], dialogue systems [65, 270, 286], image captioning [242], text summarization [343, 32, 177] and translation [236, 108]. This is especially true in low-resource settings [91]. In order for LMs to be fully utilized as such knowledge bases and in settings where factuality is crucial, the retrieved knowledge must first be factual. But, without knowing the source of such the model's knowledge, verifying its factuality is a challenge.

Citation Generation Although LMs, particularly those intended to produce scientific text, such as Meta's Galactica [285], already produce text that looks as if it is a citation, frequently there is no document corresponding to the apparent citation or the cited document does not support the statement associated with it. Many existing approaches to citation recommendation offer productive avenues to explore for factuality testing, post-hoc generation of support, hybrid architectures, or creation of training data [3, 136]. There has also been work on citation generation, where the task is either: 1) given two documents, generate an explanation for the relation between them [165], or 2) generate a citation for an already existing text [88, 317, 313, 75]. This is different from our suggestion to generate statements and citations simultaneously, and also not optimal: as LMs are being trained on massive datasets, evaluating whether each statement came from each of the potentially millions of article becomes impractical. Lastly, many existing systems that can in fact provide citations are based on search engines or retrieval models [181, 87], see also Perplexity AI[1] , YouChat[2] , or the ALCE benchmark [81]. This is problematic because 1) it is far more time consuming than directly

[1] https://www.perplexity.ai/
[2] https://you.com/

generating citations together with text; 2) access to the information sources needs to be provided at all times; 3) in contrast to our proposed approach, it does not increase model interpretability; and 4) for low-resource languages the quantity and quality of the data is often limited, and hence result in difficulties retrieving the relevant, factual source.

5.1.2 Citations and Their Pros and Cons

In this section, we will first discuss which NLP tasks – according to us – require LMs with an ability to cite their sources. We will then discuss the benefits and, subsequently, risks of such models.

5.1.2.1 Which Tasks Require Citations?

We propose to classify tasks via two questions: (1) Is the source of the generated text obvious? (2) Is the generated text an objective truth or a subjective statement? See Table 5.1 for examples.

If the answer to the first question is *yes*, no further citation is required. This is the case, e.g., for machine translation [27]: the content of the generated text comes from the input sentence. The same holds true for summarization [251] and paraphrase generation [348]. However, this is only partially the case for text simplification [267]: while most of the content comes from the original text, simpler versions of text sometimes contain additional explanations, which do require citations. In contrast, for many other tasks the input does not act as the source for text generation – instead, the output comes from information stored in the model parameters and, thus, originally from the training data. An ideal system would be able to cite the part of its training data responsible for any given output. This is the case for the popular NLP tasks of closed-book free-text question answering [240], dialogue generation [342], or creative writing [320].

For tasks for which the answer to Question 1 is *no*, we then turn to the second aforementioned question and ask if the generated text without clear sources of information in the input contains what should be objective truths. This is typically true for closed-book free-text question answering, which, as a consequence, according to our rules does require citations. However, this is *sometimes* the

case for other tasks too, such as the generation of additional explanations during text simplification or image captioning. Similarly, for dialogue generation, objective truths and subjective statements could be mixed within the same conversation. As a result, some generated statements for those tasks do require citations, while others are good without.

Task	Q1	Q2	Citation?	Example
Creative writing	No	Sometimes	Sometimes	Penguins are known for their ability to survive in harsh Antarctic conditions [CITATION], but few people know that they also possess the power of telekinesis which they use to build intricate nests out of ice blocks.
Dialogue generation	No	Sometimes	Sometimes	Did you know that penguins can jump up to 6 feet out of water when leaping onto land or ice floes? [CITATION]. I think elephants can do the same.
Free-text QA	No	Yes	Yes	The current president is not a penguin [CITATION].
Image captioning	No	Sometimes	Sometimes	A group of penguins diving into the ocean to catch fresh fish for dinner, highlighting their impressive swimming abilities [CITATION], while one penguin emerges victorious with a giant fish twice its size.
Paraphrase generation	Yes	N/A	No	**Source text**: Penguins are social animals who live in large colonies. **Paraphrased sentence**: Penguins thrive in community living
Summarization	Yes	N/A	No	**Source text**: Emperor penguins are the largest species of penguin, standing up to 4 feet tall. They are skilled hunters, capable of catching fish and krill by diving hundreds of feet below the surface. **Summary**: Emperor penguins are notable for their size and hunting prowess, making them formidable predators in their environment.
Text simplification	Sometimes	Sometimes	Sometimes	**Source text**: Penguins have evolved unique adaptations that allow them to survive in environments as harsh as Antarctica, such as their countershaded dark and white plumage, which camouflages them from predators above and below the ice. **Simplified text**: Penguins live in Antarctica, which is year-round one of the coldest places on Earth [CITATION], and they look different than other birds so they don't get eaten.
Translation	Yes	N/A	No	**Source text**: Penguins are cool. **Translated text**: Pinguine sind cool.

Table 5.1: An overview of natural language generation tasks together with our opinion regarding if they require citations. Q1: *Obvious source?* Q2: *Objective truth?*

5.1.2.2 Benefits of Citations

Citations allow us to verify the factuality of generated text easily. In contrast, without knowing where the text came from we are often unable to verify that it is correct. Moreover, knowing what portion of the text is copied verbatim allows us to give credit to the author and prevent copyright violations. Citations also increase the explainability of the answer and allow users to learn more about interesting topics.

Additionally, recent work in prompt engineering have shown that models providing justifications for their assertions (even when only partially correct) can improve the correctness of the outputs [118]. Trustworthiness judgments among people often include a social aspect, so by doing

a good job of identifying sources and influences has the potential to increase both the trust in AI systems and their trustworthiness. For example, human trustworthiness judgments about scientific claims are influenced by the interests of the authors [86].

5.1.2.3 Risks of Citations

Unfortunately, citations also come with risks. Just by having a citation next to a generated text, users are more likely to trust it [287]. However, it is likely that users will not examine each and every citation manually to verify that the text is indeed factual, or that the source is trustworthy [275, 287]. This will be exacerbated by the fact that it is incredibly unlikely that any automated system will ever produce 100% correct citations at all times, and may result in either users' diminishing trust and usage of such systems or a potential harm.

There is also the risk of decreased readability: backing up every statement with many citations, as the text may appear in multiple places, will reduce the readability of the text and may hinder users from reading or understanding it. Lastly, privacy concerns also arise from the training process of LMs. For example, state of the art LMs are often trained on a massive automatically extracted text [227]. But, as manual examination of each text is not feasible for its size, there is a possibility that it may contain private user information, such as patient records. This may result in LMs cite information that should stay private.

5.1.2.4 Citations vs. Explainability

The goal to understand why a model generates any given output is shared with research on model explainability [50]. However, in contrast to the latter, we are not interested in the effect of certain input on the output. In addition, we do not necessarily require that the model describes its reasoning by providing citations – what we care about instead is that the citations back up the model's answer. This enables humans to verify the output – even if the cited source should not actually in the technical sense have been the reason for the model's output.

5.1.3 Road Map

5.1.3.1 The Big Picture

Meta-information Currently, the standard in the field is to train models on text, disjoint from its origin. Even though some models are trained on data that contain text with citations (e.g., [285]), the citations are only "attached" to statements taken from other sources, while any other text, even taken from the same article, does not have a citation attached to it. This results in LMs that can only sometimes, on a limited text, produce citations. In order to develop LMs that can cite their sources effectively, we need to give them the metadata which contain citation information.

Retrieval Say we trained a LM with the right data such that it has knowledge of which statement came from which article. How would we extract text with citations? One avenue for such knowledge extraction is to modify the pretraining, such that citation information is being generated together with every piece of generated text.

When To Cite? The above strategy would result in LMs that would always produce a citation. However, as mentioned in Section 5.1.2.1, not every task or statement requires a citation. For tasks that do require citations, we can just let the model always cite. For tasks that do not require citations, we can simply remove the citations. For tasks in between, where citation is sometimes required, we propose to utilize the existing subjectivity classification task [311].

5.1.3.2 Concrete Tasks to Master

Our goal is to lay out a roadmap for the community, which describes necessary steps for the development of models that can cite their sources. This is not trivial, as it requires improvements of models for existing tasks as well as the development of systems for novel challenges.

Simultaneous Citation and Text Generation As mentioned in Section 5.1.1, existing work mainly retrieve citations for already generated text, which becomes intractable as models are trained on ever more text and the number of possible source documents increases drastically. In contrast, we propose STANCE: the task of **S**imultaneous **T**ext **AN**d **C**itation g**E**neration. As an

additional challenge, future work should also focus on MultiSTANCE: multihop citation generation, where the sources for a given text are spread across multiple texts. As the number of citations can be significant (though much smaller in the low-resource setting), we suggest to use topic modeling, as a potential avenue to reduce such large search space.

Subjectivity Classification As mentioned in Section 5.1.2.1, whether a task requires a citation partially depends on if the text is objective or subjective. This is not a novel task as the community has been working on subjectivity classification for quite some time [311, 310]. However, to the best of our knowledge, models for this task have not been employed in the context of citations.

Citation-Text Correctness To ensure that the retrieval step (Section 5.1.3.1) is successful, we need to identify whether the statement appears in the source. For that, two existing tasks can be used: 1) identifying which part of the generated text refers to the citation [299]. 2) Validate that the citation is appropriate for the selected text span [123, 175, 100, 185, 143]. Using such automatic methods instead of manually verifying citations will result in faster model development.

Source Trustworthiness We all know that Wikipedia is not a reliable source for citation. We propose CUE (**C**itation q**U**ality **E**valuation), the task of evaluating the quality of the source corresponding to a generated citation. We believe there are six main sub-tasks for CUE, which consist of classifying 1) the time of publication, 2) whether the source is credible, 3) how many times the source has been cited, 4) if the author is known, 5) if the source is unbiased, and 6) if the statement and citation are still relevant. For example, answering that the current US president is Barack Obama was **previously** factual, and may still show up in many source documents, but is not factual in 2023.

5.2 Adaptive Question Answering: Enhancing Language Model Proficiency for Addressing Knowledge Conflicts with Source Citations

The work described in this section has been published in EMNLP 2024 [261].

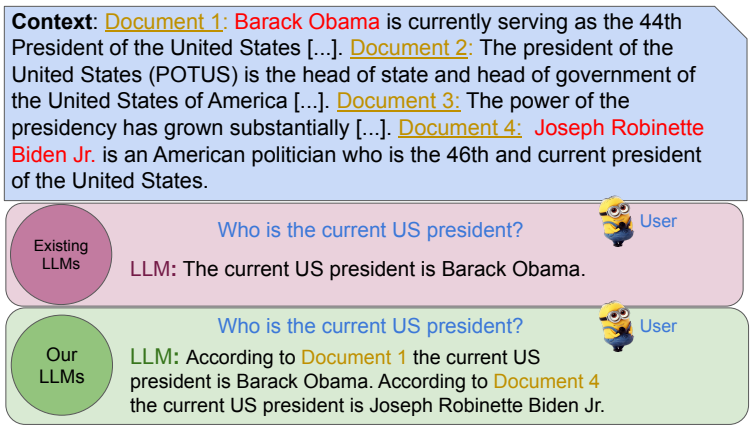


Figure 5.2: When faced with ambiguous settings, unlike existing models that often provide a single answer, our methods generate multiple answers and cite their sources, allowing users to verify the answers' factuality and make informed decisions.

5.2.1 Introduction

Knowledge-enhanced LLMs have demonstrated remarkable QA capabilities, partially due to their ability to reason over a substantial number of tokens [102, 307]. While some work has shown that LLMs do not fully utilize long sequences [157], an issue that arises *from the context itself* is that knowledge is dynamic and is constantly changing, and hence, conflicting facts and opinions may exist within it [187, 200]. For example, since politicians change, there exist documents each expressing that a different person is the current president of the United states.

This is especially problematic for models that can handle long contexts, such as existing state-of-the-art models [149, 206, 194, 206, 9] and retrieval-augmented generation (RAG) systems [146, 307, 102], as the greater the length of the context, the higher the probability of encountering conflicting information from various sources, domains, or even time periods within the same source or domain.

Existing work on QA in contextual[3] knowledge conflicts[4] setting mitigate this issue by either predicting all valid answers [188, 186], aggregating all answers [264, 82], asking clarification questions [338], and other methods [48, 280]. However, these methods burden users with the task

[3] Unlike settings where context contradicts model knowledge [200], which is not our focus.
[4] Knowledge conflicts occur when multiple answers are possible from a set of documents and a question.

of extensively evaluating the factuality of each answer [237, 257, 66], **as they do not cite the source of the answer.**

Hence, it is crucial to develop systems that not only generate all possible answers, but produce a distinct response for each conflicting information while also citing their sources, as shown in Figure 5.2. This, in turn, will also lead to an increase in users' trust and interpretability [257]. And while some existing work develop models that cite their sources, **they only focus on unambiguous setting**, where only one answer exists [24, 80, 277]. Furthermore, **none of the existing work on contextual conflicts or citation generation focus on complex QA settings**, which require multi-hop reasoning and many answers, and resemble a more realistic real-world setting [117, 191, 211]

We bridge the gap between ambiguous QA and citation generation by proposing the novel task of QA with source citation in ambiguous settings, where multiple valid answers exist. To facilitate research, we provide a comprehensive framework featuring: five novel datasets with citation metadata, the first ambiguous multi-hop QA dataset, two new evaluation metrics, and strong baselines. Our goal is to inspire the community to push the boundaries of QA research and develop more trustworthy and interpretable systems.

5.2.2 Experiments

In our experiments, each dataset consists of triples $(q, [c_1, ..., c_n], [a_1, ..., a_k])$, where q is a question, $[c_1, ..., c_n]$ are multiple context documents, and $[a_1, ..., a_k]$ are at least two conflicting answers. We follow prior work [29, 41, 258, 156] by concatenating the question and contexts into a single string, which is then input to each model.

5.2.2.1 Metrics

To comprehensively assess the performance of systems tackling the novel task of QA with source citation in ambiguous settings, we introduce two novel evaluation metrics that capture the ability to generate distinct responses for conflicting information while accurately citing sources.

Specifically, for each question q, we evaluate the generated response along two crucial dimensions:

Acc_K: This metric measures the ability to produce a diverse set of correct answers, with a focus on generating at least K of the gold answers. For instance, if the gold answers are ["X", "Y", "Z"] and the generated answers are ["X", "Y"], the scores would be: Acc_1=1, Acc_2=1, Acc_3=0.

Citation Accuracy (A_C): This metric assesses the ability to accurately generate citation strings corresponding to the correct sources. For example, if the gold answers are ["According to Document X the answer is X1", "According to Document Y the answer is Y1"] and the generated answers are ["According to Document X the answer is X1", "According to Document Z the answer is Y1"], the score would be 0.5.

By utilizing these two metrics, we can gain a more nuanced understanding of system performance in resolving knowledge conflicts and citing sources accurately, ultimately driving progress in this critical task. For both accuracy measures, we follow [157, 171, 121] and evaluate if the gold answer or citation string are present in the output.

5.2.2.2 Datasets

Notably, existing QA datasets lack citation metadata, which is a critical component of our proposed task. To address this gap, we augment three reading comprehension (RC) datasets to create novel evaluation sets[5] that focus on different conflicting settings, each enriched with citation metadata. Specifically, we add a unique citation string "Document X" before each document context c_i, where X represents a distinct document identifier (as illustrated in Figure 5.2). In real-world scenarios, these citation strings can correspond to PubMed IDs, Wikipedia IDs, or other types of document identifiers. To further increase the task's complexity and realism, we add citation strings before each paragraph in longer contexts, such as multi-hop settings. This design choice presents a dual benefit: models must now reason through and produce multiple citations, while users can more easily identify relevant information without having to parse entire documents.

[5] Which we will make publicly available.

AmbigQA-Cite We build upon AmbigQA [187], an open-domain RC dataset, which is derived from the Natural Questions (NQ) dataset [138] and comprises 14,042 questions. Notably, AmbigQA-Cite features *ambiguous questions*. To create our citation-augmented dataset, we employ the following methodology: for each ambiguous question, we select contexts that contain exactly one of the answers and exclude those that contain multiple answers. We further restrict our dataset to questions with exactly two conflicting answers, each supported by a distinct context, as questions with more conflicting answers are extremely rare and would lead to limited sample sizes and unreliable conclusions. The resulting dataset, which we term *AmbigQA-Cite*, is enriched with citation information to support the development of more accurate and trustworthy question answering models.

DisentQA-DupliCite

We use DisentQA [200], an open-domain RC QA dataset with 108,291 questions from the NQ dataset. Unlike AmbigQA, DisentQA focuses on *ambiguous contexts*, where the question is clear, but the answer varies depending on the context (Figure 5.2). The dataset uses entity-substitution [163] to create conflicting contexts, resulting in 39,716 pairs of questions with two conflicting contexts and answers each. Notably, this substitution approach leads to context duplication, where both contexts for each question are similar except for the replaced entity. We augment this dataset with citation information, creating *DisentQA-DupliCite*.

DisentQA-ParaCite To mitigate the potential shortcut issue in *DisentQA-DupliCite*, where models may exploit the similarity between duplicated contexts, we create a paraphrased version of each conflicting context for each question. Specifically, we use ChatGPT [206] to paraphrase each conflicting context, taking care to preserve the replaced entity in the output using the specific prompt: "*Paraphrase this: {conflicting_context}. Ensure that {conflicting_label} is still in the paraphrased output*". This process yields a new dataset, which we term *DisentQA-ParaCite*, featuring paraphrased contexts that require models to engage in more robust and meaningful reasoning.[6]

[6] We manually evaluate 100 paraphrased examples and found that 98% were of high quality.

Conflicting HotPotQA-Cite HotPotQA [324] is a multi-hop RC QA dataset with 112,779 questions We use a masked language model (MLM) approach, similar to [259, 210, 152], to introduce conflicting contexts. We opt for MLM over entity substitution to preserve text grammatical integrity [67]. Using DistilBERT [247], we generate two conflicting answers per context, creating three conflicting answers and contexts per question. This yields the *Conflicting HotPotQA-Cite* dataset, the first conflicting multi-hop QA dataset with real-world, naturally occurring contexts. Unlike BoardgameQA [130], our dataset features complex contextualized contradictions.

We provide two variants of this dataset: (1) a *with distractors* version, which includes up to 14 cited documents in each context, including both relevant and distracting contexts, and (2) a *no distractors* version, which only includes the relevant contexts, limited to up to 6 cited documents.

5.2.2.3 Models

Model	Paramaters	Training Dataset Size
Llama-7B Chat	7 billion	2 trillion tokens
Llama-13B Chat	13 billion	2 trillion tokens
Llama-70B Chat	70 billion	2 trillion tokens
MPT-7B Instruct	7 billion	1 trillion tokens
Falcon-7B	7 billion	1.5 trillion tokens

Table 5.2: Models, their size, and the number of tokens in their training data.

We experiment with 5 different LLMs: Llama-2-7B Chat [290], Llama-2-13B Chat [290], Llama-2-70B Chat[7] Instruct [290], MPT-7B [194], and Falcon-7B Instruct [5]. A summary can be seen in Table 5.2.

5.2.2.4 Baselines

In addition to introducing the novel task of QA with source citation in ambiguous settings, we establish a set of strong baseline models to facilitate progress in this area. Our proposed baselines comprise a range of approaches, including rule-based, prompting-based, and finetuning-based models.

[7] We evaluate the 70B model on most settings, except a few due to unexpected computational constraints.

In the following examples q_{e1},...,q_{ek} and c_{e1},...,c_{en} are in-context learning question and contexts; q_{ti} is the test question and c_{ti} are the test contexts. The citations are included in the contexts.

Zero-shot Baseline We concatenate the question and contexts into a single input string, as described in Section 5.2.2, and feed it to each of the models.

5.2.2.5 Prompt-based Methods

Conflict-aware Basic Prompting We employ a few-shot approach [29] with 1, 3, or 5 examples per prompt, utilizing a structured prompt design that explicitly acknowledges the presence of conflicting information. This conflict-aware (C.A) prompting design emphasizes the existence of conflicting information and its corresponding citations, enabling models to develop a more nuanced understanding of ambiguous contexts.

Few-shot Conflict-aware CoT Prompting We adopt the few-shot Chain-of-Thought (CoT) method [306], which involves providing the model with explicit reasoning steps to arrive at an answer. We create 1 or 3 manually-crafted CoT examples that highlight conflicting information and their associated citations, and append them to the prompt, enabling the model to generate an answer in a single step.

Zero-shot CoT Prompting In the zero-shot approach [134], we employ a two-step process to elicit reasoning from the model:

Unlike the C.A CoT method, here, we do not provide explicit examples of conflicting context and citations. Instead, we aim to assess whether the model's self-generated reasoning paths are sufficient to handle conflicting facts.

5.2.2.6 Rule-based Methods

Document Split

Our rule-based approach, *Document Split*, employs a predetermined set of rules to process the context. Specifically, we split the context into individual articles based on the citation tokens, and process them sequentially, following a strict rule: each article is processed one at a time, rather than

all at once. This approach makes citations trivial, as we can generate one response per document and evaluate them separately to identify correct citations. However, this rule-based approach also has a limitation. Since models can only see one document at a time, they are incapable of answering questions that require complex reasoning across multiple documents.

5.2.2.7 Finetuning Methods

Fine-tuning with Low-Rank Adaptation (LoRA) We fine-tune LLMs on our datasets using LoRA [101], a parameter-efficient technique that avoids full model fine-tuning. LoRA adds small, trainable adapters to specific layers, keeping original parameters frozen, and allows control over adapter influence via the alpha value. We fine-tune each model on each of the following datasets: AmbigQA-Cite, DisentQA-DupliCite, and Conflicting HotPotQA-Cite (without distractors).

5.2.3 Results

5.2.3.1 Ambiguous Questions

Method / Model	Zero-Shot	C.A Basic Prompting			Few-shot C.A CoT		Zero-shot CoT	Document Split	Finetuning
		1-shot	3-shot	5-shot	1-shot	3-shot	1-shot		
Llama-7B	A_1: 54.8 A_2: 2.1 A_C: 0.0	A_1: 54.8 A_2: 20.4 A_C: 33.3	A_1: 62.3 A_2: 21.5 A_C: 34.4	A_1: 61.2 A_2: 24.7 A_C: 34.9	A_1: 41.9 A_2: 8.6 A_C: 2.1	A_1: 56.9 A_2: 13.9 A_C: 0.5	A_1: 45.1 A_2: 2.1 A_C: 0.0	A_1: 67.7 A_2: 30.1 A_C: NA	A_1: **69.8** A_2: **35.4** A_C: **48.3**
Llama-13B	A_1: 48.3 A_2: 3.2 A_C: 0.0	A_1: 63.4 A_2: 23.6 **A_C: 36.5**	**A_1: 67.7** A_2: 22.5 **A_C: 36.5**	A_1: 63.4 A_2: 23.6 A_C: 34.9	A_1: 61.2 A_2: 10.7 A_C: 5.9	A_1: 55.9 A_2: 15.0 A_C: 9.6	A_1: 45.1 A_2: 1.0 A_C: 0.0	A_1: 62.3 A_2: 21.5 A_C: NA	A_1: 66.6 A_2: **32.5** A_C: 30.6
Llama-70B	A_1: 54.8 A_2: 4.3 A_C: 0.0	A_1: 72.0 **A_2: 35.4** A_C: 45.6	A_1: 74.1 **A_2: 35.4** **A_C: 48.3**	A_1: 70.9 A_2: 30.1 A_C: 45.6	A_1: 70.9 A_2: 30.1 A_C: 29.0	A_1: 73.1 A_2: 31.1 A_C: 31.7	A_1: 38.7 A_2: 4.3 A_C: 0.0	**A_1: 76.3** A_2: 25.8 A_C: NA	A_1: - A_2: - A_C: -
MPT-7B	A_1: 50.5 A_2: 0.0 A_C: 0.0	A_1: 51.6 A_2: 9.6 A_C: 12.9	A_1: 46.2 A_2: 9.6 **A_C: 21.5**	A_1: 44.0 A_2: 7.5 A_C: 19.8	A_1: 47.3 A_2: 3.2 A_C: 8.6	A_1: 45.1 A_2: 2.1 A_C: 6.9	A_1: 45.1 A_2: 1.0 A_C: 0.0	**A_1: 65.5** **A_2: 21.5** A_C: NA	A_1: 51.6 A_2: 10.7 A_C: 16.1
Falcon-7B	A_1: 30.1 A_2: 1.0 A_C: 0.0	A_1: 8.6 A_2: 2.1 A_C: 4.8	A_1: 39.7 A_2: 5.3 **A_C: 16.6**	A_1: 25.8 A_2: 4.2 A_C: 8.0	A_1: 26.8 A_2: 3.2 A_C: 11.2	A_1: 36.5 A_2: 3.2 A_C: 13.9	A_1: 30.1 A_2: 1.0 A_C: 0.0	**A_1: 52.6** A_2: 9.6 A_C: NA	A_1: 48.3 A_2: **19.3** A_C: 13.9

Table 5.3: AmbigQA-Cite Results. Accuracy scores are reported as percentages. The Document Split method involves providing each document individually to the models, and hence, citations are known by default. C.A=Conflict-aware.

We first analyze the ability of models to answer ambiguous questions on the AmbigQA-Cite dataset Results can be seen in Table 5.3.

Our analysis of the zero-shot baselines reveals that most models can answer at least one of the two answers correctly (A_1) around 50% of the time, with Llama-70B performing the best

at 54.8% and Falcon-7B performing the worst at 30.1%. However, all models struggle to produce distinct answers, with the best A_2 score being 4.3% for Llama-70B. Moreover, none of the models generate citations, resulting in 0% A_C across all models.

The various prompting methods show improvement in models' ability to answer at least one answer correctly (A_1), with the best method – C.A basic – yielding the highest increase in performance, on Llama-13B with a 19.4% increase. Almost all methods, except for the zero-shot CoT, also improve models' ability to generate distinct responses, with the finetuning method showing the highest increase in A_2 accuracy, on Llama-7B with a 33.3% increase. In contrast, the zero-shot CoT method performs poorly, with most models and metrics showing a decrease in performance.

The document split method improves all models' A_1 scores, but not always their A_2 scores, where finetuning results are mixed, with some models (like Llama-7B) outperforming the best prompting method, while others (like Llama-70B) show comparable or weaker performance.

5.2.3.2 Ambiguous Context: Single-hop

Method / Model	Zero-Shot	C.A Basic Prompting			Few-shot C.A CoT		Zero-shot CoT	Document Split	Finetuning
		1-shot	3-shot	5-shot	1-shot	3-shot	1-shot		
Llama-7B	A_1: 84.6 A_2: 10.2 A_C: 0.0	A_1: 85.9 A_2: 64.0 A_C: 51.6	**A_1: 88.5** **A_2: 76.4** **A_C: 77.6**	A_1: 0.1 A_2: 0.0 A_C: 0.0	A_1: 81.7 A_2: 51.4 A_C: 14.4	A_1: 86.3 A_2: 68.7 A_C: 50.0	A_1: 81.5 A_2: 14.9 A_C: 0.0	A_1: 87.5 A_2: 49.0 A_C: NA	A_1: 79.3 A_2: 61.0 A_C: 58.5
Llama-13B	A_1: 82.2 A_2: 10.5 A_C: 0.0	A_1: 89.0 A_2: 74.5 A_C: 76.0	**A_1: 91.9** **A_2: 79.0** **A_C: 81.9**	A_1: 0.1 A_2: 0.0 A_C: 0.0	A_1: 86.8 A_2: 55.0 A_C: 23.6	A_1: 90.2 A_2: 74.3 A_C: 45.5	A_1: 80.7 A_2: 9.8 A_C: 0.0	A_1: 85.9 A_2: 40.0 A_C: NA	A_1: 81.6 A_2: 68.0 A_C: 68.1
Llama-70B	A_1: 88.3 A_2: 16.4 A_C: 0.0	A_1: 93.6 A_2: 85.1 A_C: 76.4	**A_1: 94.1** **A_2: 88.3** **A_C: 86.7**	A_1: 0.1 A_2: 0.0 A_C: 0.0	A_1: 92.3 A_2: 66.7 A_C: 26.7	A_1: 93.4 A_2: 83.6 A_C: 52.5	A_1: 75.8 A_2: 16.8 A_C: 0.0	A_1: 91.5 A_2: 45.8 A_C: NA	A_1: - A_2: - A_C: -
MPT-7B	A_1: 80.3 A_2: 2.7 A_C: 0.0	A_1: 78.2 A_2: 42.3 A_C: 43.1	A_1: 74.0 A_2: 49.3 **A_C: 54.1**	A_1: 0.1 A_2: 0.0 A_C: 0.0	A_1: 75.7 A_2: 35.7 A_C: 13.9	A_1: 70.9 A_2: 30.0 A_C: 10.0	A_1: 77.6 A_2: 4.8 A_C: 0.0	**A_1: 82.7** **A_2: 56.6** A_C: NA	A_1: 61.0 A_2: 21.0 A_C: 9.6
Falcon-7B	A_1: 63.2 A_2: 16.6 A_C: 0.0	A_1: 50.6 A_2: 35.6 A_C: 37.5	A_1: 70.3 **A_2: 45.8** **A_C: 53.0**	A_1: 0.0 A_2: 0.0 A_C: 0.0	A_1: 54.7 A_2: 30.0 A_C: 27.1	A_1: 69.9 A_2: 44.0 A_C: 42.0	A_1: 61.3 A_2: 10.8 A_C: 0.0	**A_1: 71.8** A_2: 38.3 A_C: NA	A_1: 71.6 A_2: 45.6 A_C: 46.3

Table 5.4: DisentQA-DupliCite Results. The Document Split method involves providing each document individually to the models, and hence, citations are known by default.

We next analyze the ability of models to answer questions with ambiguous contexts on the DisentQA-DupliCite and DisentQA-ParaCite datasets. Results can be seen in Tables 5.4 and 5.5.

Out-of-the-box models are unable to generate citations, and generally struggle to produce

Method / Model	Zero-Shot	C.A Basic	Few-shot C.A CoT	Finetuning
Llama-7B	A_1: 69.6 A_2: 7.3 A_C: 0.0	A_1: 74.3 A_2: **56.0** A_C: **59.0**	A_1: 65.3 A_2: 47.6 A_C: 35.0	A_1: **74.6** A_2: 54.0 A_C: 40.0
Llama-13B	A_1: 71.6 A_2: 4.3 A_C: 0.0	A_1: 77.0 A_2: 58.0 A_C: 60.6	A_1: 72.0 A_2: 40.6 A_C: 10.8	A_1: **81.6** A_2: **66.6** A_C: **67.1**
MPT-7B	A_1: **65.3** A_2: 0.3 A_C: 0.0	A_1: 54.3 A_2: **26.6** A_C: 32.0	A_1: 56.0 A_2: 14.3 A_C: 4.3	A_1: 64.6 A_2: 22.6 A_C: **10.8**
Falcon-7B	A_1: 50.6 A_2: 7.6 A_C: 0.0	A_1: 54.3 A_2: 19.3 A_C: 30.3	A_1: 58.3 A_2: 22.0 A_C: 30.1	A_1: **69.3** A_2: **41.6** A_C: **40.1**

Table 5.5: DisentQA-ParaCite. C.A=Conflict-aware. We use 3 examples for both C.A Basic and CoT.

multiple answers, resulting in poor A_2 scores. Most methods improve models' A_1 scores and their ability to generate distinct responses, with the best prompting method being C.A basic using 3 in-context examples. In contrast, the Zero-shot CoT method performs poorly. We also find that with 5 examples, the performance on DisentQA-DupliCite drops due to context size exceeding the models' maximum capacity, leading to test question truncation.

Notably, models' scores are significantly higher on the DisentQA-DupliCite dataset, with A_1 scores ranging from 70.3% to 94.1% using the C.A basic method (3-shot), compared to 39.7% to 76.2% on AmbigQA-Cite. The document split method improves all models' performances, but only outperforms the few-shot method for MPT-7B and Falcon-7B models on A_1.

In contrast, the DisentQA-ParaCite dataset presents a more challenging scenario, with overall lower scores than on DisentQA-DupliCite. However, we observe similar behavior, with C.A basic and finetuning methods yielding comparable scores. Interestingly, finetuning emerges as the overall best method on DisentQA-ParaCite.

5.2.3.3 Ambiguous Context: Multi-hop

We evaluate our baselines on the more complex Conflicting HotpotQA-Cite dataset, which involves multi-hop QA with many conflicting answers. The results are presented in Tables 5.6 and

Method/ Model	Zero-Shot	C.A Basic	Few-shot C.A CoT	Finetuning
Llama-7B	A_1: 82.6 A_2: 27.6 A_3: 5.0 A_C: 0.0	A_1: 67.0 A_2: 36.1 A_3: 10.3 A_C: 8.9	A_1: 75.0 A_2: 25.0 A_3: 10.0 A_C: 11.6	A_1: **98.0** A_2: **90.0** A_3: **62.0** A_C: **67.3**
Llama-13B	A_1: 83.0 A_2: 21.3 A_3: 4.6 A_C: 0.0	A_1: 86.0 A_2: 68.9 A_3: 37.0 A_C: 36.3	A_1: 80.0 A_2: 55.0 A_3: 25.0 A_C: 13.3	A_1: **98.3** A_2: **93.3** A_3: **65.6** A_C: **76.3**
MPT-7B	A_1: 72.3 A_2: 16.0 A_3: 2.0 A_C: 0.0	A_1: 65.3 A_2: 24.0 A_3: 4.8 A_C: 0.03	A_1: 50.0 A_2: 15.0 A_3: 0.0 A_C: 0.0	A_1: **93.0** A_2: **84.3** A_3: **59.0** A_C: **64.5**
Falcon-7B	A_1: 63.0 A_2: 24.6 A_3: 6.3 A_C: 0.0	A_1: 48.1 A_2: 15.4 A_3: 2.6 A_C: 0.01	A_1: 0.0 A_2: 0.0 A_3: 0.0 A_C: 0.0	A_1: **85.3** A_2: **75.3** A_3: **39.3** A_C: **49.7**

Table 5.6: Conflicting HotpotQA-Cite (no distractors). C.A=Conflict-aware. We use 3 examples for both C.A Basic and CoT.

Method/ Model	Zero-Shot	C.A Basic	Few-shot C.A CoT	Finetuning
Llama-7B	A_1: **59.8** A_2: 16.8 A_3: 2.2 A_C: 0.0	A_1: 38.0 A_2: 9.3 A_3: 1.0 A_C: 0.1	A_1: 39.3 A_2: 12.6 A_3: 0.6 A_C: 0.2	A_1: 49.0 A_2: **17.3** A_3: **2.3** A_C: **2.0**
Llama-13B	A_1: **60.8** A_2: 15.8 A_3: 3.5 A_C: 0.0	A_1: 51.0 A_2: **20.0** A_3: **2.6** A_C: **1.3**	A_1: 42.3 A_2: 13.0 A_3: 2.3 A_C: 0.1	A_1: 46.6 A_2: 16.3 A_3: 2.0 A_C: 1.2
MPT-7B	A_1: **49.5** A_2: 13.0 A_3: **3.0** A_C: 0.0	A_1: 31.0 A_2: 8.6 A_3: 1.0 A_C: 0.0	A_1: 42.0 A_2: 12.3 A_3: 1.3 A_C: 0.0	A_1: 48.6 A_2: **15.3** A_3: 1.6 A_C: **0.4**
Falcon-7B	A_1: 29.5 A_2: 7.5 A_3: 2.5 A_C: 0.0	A_1: 25.3 A_2: 8.0 A_3: 1.3 A_C: 0.0	A_1: 4.3 A_2: 0.0 A_3: 0.0 A_C: 0.0	A_1: **37.4** A_2: **8.4** A_3: **8.6** A_C: **0.2**

Table 5.7: Conflicting HotpotQA-Cite (Distractors). C.A=Conflict-aware. We use 3 examples for both C.A Basic and CoT.

5.7.

On the no-distractor variant dataset, we observe two unexpected trends. While the C.A basic method improves models' performances on A_2, A_3, and A_C metrics, it underperforms

the zero-shot baseline on A_1. In contrast, finetuning significantly outperforms all other methods, achieving nearly 100% A_1 scores across Llama-7B, Llama-13B, and MPT-7B. However, all models are still far from perfect on generating all correct answers correctly, in addition to citing their sources. Lastly, the few-shot CoT method generally performs poorly across all models and metrics.

On the distractor variant dataset, the C.A basic method underperforms the zero-shot baseline on A_1, but outperforms it on A_2, A_3, and A_C. The overall models' scores are significantly lower than on the no-distractor setting, indicating that this setting is more challenging for models. Finetuning again emerges as the best approach, outperforming most methods. However, all baselines struggle to generate multiple correct answers, with the best scores being 17.3% for A_2 (finetuned Llama-7B) and 8.6% for A_3 (finetuned Falcon-7B). Additionally, they perform poorly on citing their sources.

5.2.3.4 Non-ambiguous Context: Single-hop

Method/ Model	**Zero-Shot**	**C.A Basic**	**Finetuning**
Llama-7B	**A_1: 84.8**	A_1: 84.1	A_1: 64.3
Llama-13B	**A_1: 83.8**	A_1: 80.3	A_1: 71.6
Llama-70B	**A_1: 89.0**	A_1: 88.1	A_1: -
MPT-7B	**A_1: 81.2**	A_1: 73.9	A_1: 51.3
Falcon-7B	A_1: 71.1	**A_1: 72.5**	A_1: 46.3

Table 5.8: DisentQA with no contextual conflicts. C.A=Conflict-aware. We use 3 examples for C.A Basic.

We assess whether the top-performing techniques, C.A basic and finetuning, degrade models' performances compared to the zero-shot baseline when no ambiguity exists. We use the original context from the DisentQA dataset, which lacks knowledge conflicts. The results are presented in Table 5.8.

For the C.A basic method, we observe that most models experience some performance degradation, except for Falcon-7B, which actually shows a performance increase. For example, MPT-7B suffers the largest A_1 drop, from 81.2% to 73.9%, while Llama-7B experiences the smallest drop,

from 84.8% to 84.1%. However, this performance drop is relatively small compared to the significant gains provided by this method in Sections 5.2.3.1, 5.2.3.2, and 5.2.3.3. In contrast, finetuning results in a much more substantial performance drop. For instance, Falcon-7B's A_1 score plummet from 71.1% to 46.3%.

5.2.4 Discussion

5.2.4.1 Ambiguous Questions vs. Contexts

We observe a significant performance gap between the AmbigQA-Cite and DisentQA-DupliCite datasets. This disparity can be attributed to two primary factors. 1) DisentQA-DupliCite is constructed using the entity-substitution method, which generates two contexts with a single differing entity answer. This design makes the task relatively easier compared to AmbigQA-Cite, where no duplicates exist. 2) AmbigQA-Cite's questions are intentionally ambiguous, rendering them more challenging to answer than those in DisentQA-DupliCite. Moreover, we observe that models perform worse on DisentQA-ParaCite, suggesting that paraphrased contexts introduce a higher level of complexity compared to entity substitution, which helps to bridge the performance gap.

5.2.4.2 Multi-hop vs. Single Hop

DisentQA-DupliCite and conflicting HotpotQA datasets share a common approach to creating conflicting contexts: replacing the answer string with a different string, yielding duplicated content. Comparing the results in Tables 5.4 and 5.6, we observe two significant trends: firstly, generating correct citations is much more challenging in the multi-hop setting, where multiple documents exist and are required to reach the answer. Secondly, producing all correct answers is also much harder, even with a limited number of correct ones. Moreover, the presence of distractors in the conflicting HotpotQA dataset further exacerbates this challenge, leading to an even more significant performance drop. These results underscore the importance of developing novel conflicting multi-hop QA datasets.

5.2.4.3 3-shot vs. 5-shot

While on the AmbigQA dataset we see a drop in performance across all models between the 3-shot and 5-shot few-shot method (see Table 5.3), the performance drop is far more significant in Table 5.4 on the DisentQA dataset. Analyzing this further, we find that with 5 examples the context becomes larger (especially on the DisentQA dataset) than the maximum context length the models can handle, which results in the test question truncation.

5.2.4.4 C.A Prompting vs. Zero-shot CoT

One possible reason for the zero-shot CoT's poor performance on A_C, with a score of 0% across all models and tested datasets, is that it lacks an explicit citation prompt. Unlike the C.A methods, which specifically ask models to cite their sources, the zero-shot method only generates a reasoning chain in the first step, without explicitly requesting citation. This highlights the necessity of a specific citation prompt. Furthermore, we observe a significant difference in A_2 scores between the two methods in both Tables 5.3 and 5.4, suggesting that models' self-generated reasoning chains are insufficient to handle conflicting facts.

5.2.4.5 Limited Efficacy of C.A. Prompts on HotpotQA

We find that both the C.A basic and C.A CoT perform worse than the zero-shot baseline and finetuning approach on the conflicting HotpotQA-Cite datasets. We hypothesize that this may be due to several reasons, such as the complexity of the multi-hop contexts, more cited documents in the multi-hop dataset, or that the in-context examples in the multi-hop setting were not as beneficial.

5.2.4.6 Finetuning vs. Prompting

Consistent with prior work [29, 158, 306, 61], our results show that the C.A prompting method can achieve comparable or even better performance than finetuning on AmbigQA-Cite, DisentQA-ParaCite, and DisentQA-DupliCite. However, it struggles on Conflicting HotpotQA-Cite. Notably,

finetuned models experience significant degradation when no conflicts exist. Overall, we conclude that the C.A basic method is the most effective approach, but both methods have room for improvement (see Section 6).

5.2.5 Real-world Usage

In our comprehensive analysis, we evaluate three main approaches to improve LLMs' ability to answer ambiguous questions with source citations: 1) prompt-based; 2) rule-based, and 3) fine-tuning-based. Notably, while the rule-based approach outperforms the other two in some occasions, as discussed in Section 5.2.2.6, it is incapable of answering questions that require complex reasoning across multiple documents, as it only sees one document at a time. To that end, we do not recommend using this approach when it is known that the data is of complex nature. But, to use it, users need to split the retrieved documents into chunks of one document at a time, which are sent to the model, followed by an aggregation of the answers. With regards to the other two approaches, the prompt-based approaches can be incorporated into most LLMs with a simple addition of a prompt, as shown in Appendix **??**. However, it is worth mentioning that the fine-tuning approach outperforms the prompting approach on multihop reasoning, but also results in a large performance decrease when no ambiguity exists, as discussed in Section 5.2.3.4. We also showed that LoRA-based fine-tuning is sufficient to improve LLMs' abilities in this task greatly over the baseline, highlighting the usability for real-users that do not have large computational resources.

5.3 More Experts Than Galaxies: Conditionally-overlapping Experts With Biologically-inspired Fixed Routing

The work described in this section has been published in ICLR 2025 [262].

5.3.1 Introduction

In recent years, there has been a trend towards developing increasingly larger models [206, 207, 73, 271, 43], driven by the understanding that a neural network's learning capacity depends

on its number of parameters [266]. This approach has yielded impressive results in various fields, including computer vision [62, 133] and language modeling [207, 43]. However, with such large size come difficulties, including increased training costs

and growing requirements for large amounts of memory and storage.

One approach to mitigating some of these challenges is sparsity, where a subset of the model's parameters is selectively utilized in the computational graph. This concept of sparsity has been widely explored in machine learning [105, 116, 142, 347, 98, 266, 16]. Researchers have observed significant benefits of sparsity, including reduced inference costs [95, 266], improved generalization capabilities [142, 106, 76, 216], enhanced learning efficiency [142], accelerated learning speed [142, 190], less interference and forgetting [231], forward knowledge transfer [8, 231, 331], and compositionality [221].

Early work on sparsity in neural networks focused on simple methods such as Dropout [279] and L1 regularization [288, 202]. These and subsequent works explore sparsity at a fine level of granularity, including single parameters [172, 173], individual neurons [318], or CNN filters [36]. Other research has explored sparsity at the level of whole networks or sub-networks within the mixture of experts (MoE) framework [266, 22]. These methods generally utilize a routing or gating function [244, 243, 266, 221], which decides which parameters or sub-networks of the model to activate based on the input.

However, existing sparse methods have limitations, which we would summarize as five key concerns: Firstly, most approaches rely on trainable gating functions [266, 193, 235, 148, 230, 36, 265, 350, 92, 154, 13, 21, 73, 172, 173, 74, 131]. This design choice is problematic for several reasons, including forgetfulness in continual learning [221, 231], representation collapse [i.e., degenerate experts; 35, 221], complex training procedures [243], and other issues [244, 221, 243]. Moreover, using non-trainable routing functions can be more effective [190, 196]. Secondly, many state-of-the-art systems employ architectures based on disjoint experts that do not share parameters [266, 221]. This design choice can lead to redundancies and may limit generalization; overlap can also be beneficial [77, 170]. Thirdly, even when experts overlap, it is unclear whether models can effectively learn to

map similar inputs to the same experts, potentially resulting in redundancies [35] or interference [221]. Fourthly, many existing methods require input or task IDs to determine which mask to apply [172, 322, 176, 170, 218, 190, 196, 122, 312], which can be restrictive, as meta-information about inputs is rarely available in real-world applications [4, 329, 302]. Lastly, the number of experts in current systems is limited, often ranging from a few to a couple of thousand [266, 110], which may not be sufficient for complex tasks [235].

In this paper we introduce Conditionally Overlapping Mixture of ExperTs (COMET), a general deep learning method that induces a modular, sparse architecture in neural networks, with a number of important properties. First, COMET uses a non-trainable gating function, eliminating the need for iterative pruning or continuous sparsification. Instead, we employ a fixed random projection followed by a k-winner-take-all cap operation, inspired by the brain's efficient use of a limited number of active cells via lateral inhibition. As in the brain, these mechanisms combine to produce sparse representations with overlap that depends on input similarity [31]. Second, COMET does not require fixed specialization of sub-networks or advance knowledge of the active neurons required for each task, enabling more flexibility and adaptability. Third, the number of possible experts in COMET is exponential in the model size, exceeding the limit of a few thousand in recent work, to effectively tackle more complex tasks. Fourth, these experts overlap based on unsupervised information from input similarities. This yields faster learning and improved generalization. It does this without increasing the number of trainable parameters, or requiring input or task IDs to determine which mask to apply.

COMET integrates concepts from diverse research areas into a concise framework: fixed random projection and k-winner-take-all from neuroscience, routing functions from modular neural networks, expert-based approaches from the MoE literature, the notion of implicit experts from dynamic neural networks, the integration of sparsity and modularity from conditional computation, input-dependent masking from various deep learning areas, and the importance of active parameter overlap from continual learning. The present paper focuses on learning and out-of-sample generalization in single tasks, but we conjecture COMET's input-dependent sparsity will also yield

advantages for settings involving multiple tasks, including transfer learning, continual learning, and robustness to catastrophic forgetting.

We validate our approach through experiments on seven diverse tasks, including image classification, language modeling, and regression, demonstrating that our method is applicable to many popular model architectures such as vision transformers, MLP-mixers, GPTs, and standard MLPs, and consistently provides improved performance. Our code will be linked here upon paper acceptance.

5.3.2 Mixture of Experts and Input-dependent Masking

A standard MoE architecture involves a disjoint set of experts and a gate that combines their predictions [105]. For example in the sparse MoE framework proposed by [266], each MoE module consists of n expert networks, $E_1, ..., E_n$, and a gating network, G, that outputs a sparse n-dimensional vector of mixture weights. The gating and expert networks are all trainable, each with its own set of parameters. The prediction for an input $\boldsymbol{x}$ is $\sum_{i=1}^{n} G(\boldsymbol{x})_i E_i(\boldsymbol{x})$.

Separately, several recent works have proposed versions of **input-dependent masking** [170, 172, 173, 322]. The general framework involves a network with n neurons and a masking function $m\colon \mathcal{X} \to \{0,1\}^n$ (where $\mathcal{X}$ is the input space). In processing an example $\boldsymbol{x}$, the network's activations are multiplied elementwise with $m(\boldsymbol{x})$. Thus the prediction for $\boldsymbol{x}$ is $F_{m(\boldsymbol{x})}(\boldsymbol{x})$ where $F_{m(\boldsymbol{x})}$ is the function computed by the sub-network corresponding to the mask $m(\boldsymbol{x})$.

Combining these two lines of work, we propose to view the sub-networks defined by input-dependent masking as **overlapping experts**. Any two experts will typically share many active neurons, and hence weights. This is in contrast to standard MoE where the experts learn disjoint sets of parameters. In the overlapping MoE framework, every subset of the full network is a (potential) expert. The sub-network $F_{m(\boldsymbol{x})}$ is an expert for $\boldsymbol{x}$, and it is also a partial expert for any other $\boldsymbol{x}'$ to the degree that $m(\boldsymbol{x})$ and $m(\boldsymbol{x}')$ overlap, as determined by the inner product $m(\boldsymbol{x})^\top m(\boldsymbol{x}')$. In the overlapping MoE framework, the gating network G is replaced by the masking function m.

We further propose that similar inputs should map to similar (i.e., more overlapping) experts.

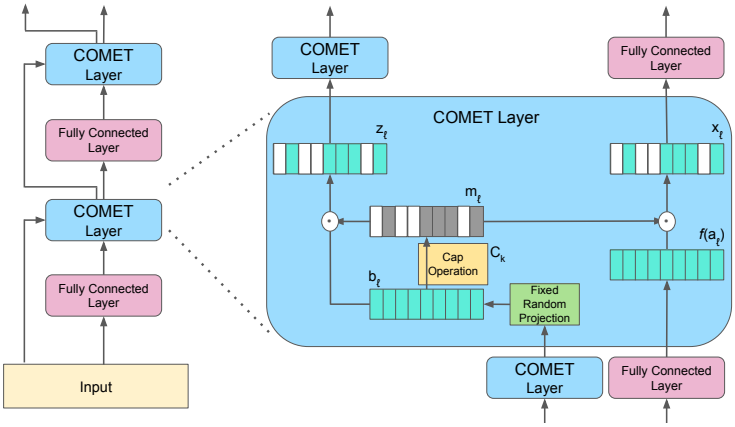


Figure 5.3: Illustration of a 2-layer MLP with embedded COMET layers. Note that COMET layers do not contain predefined experts, but instead dynamically selects a subset of the backbone MLP's parameters to activate, effectively creating implicit experts. The sparsity level determines the proportion of parameters to activate. Real value in teal, zeros in white, ones in grey.

This will facilitate generalization because what is learned about one input will selectively generalize to similar inputs. The next section explains how COMET achieves this property using biologically inspired fixed random projections ($\boldsymbol{V}_\ell$ in 5.4) and k-winner-take-all capping (C_{k_ℓ} in 5.5).

5.3.3 Conditionally Overlapping Mixture of Experts (COMET)

Our proposed COMET method applies to any backbone NN, augmenting it with a second NN called a routing network that computes input-dependent masks for all layers of the backbone network.

We first describe the COMET architecture for the case where the backbone network is an MLP. Let the backbone MLP have L layers, with layer ℓ having N_ℓ neurons and learnable parameters comprising a weight matrix $\boldsymbol{W}_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and bias $\boldsymbol{b}_\ell \in \mathbb{R}^{N_\ell}$. Then the forward pass of the unmodified MLP is defined by

$$\boldsymbol{a}_\ell = \boldsymbol{W}_\ell \boldsymbol{x}_{\ell-1} + \boldsymbol{b}_\ell \tag{5.1}$$

$$\boldsymbol{x}_\ell = f(\boldsymbol{a}_\ell) \tag{5.2}$$

for $1 \leq \ell \leq L$, where $\boldsymbol{a}_\ell$ is the pre-activation at layer ℓ, f is the elementwise activation function, $\boldsymbol{x}_0$ is the input to the network, and $\boldsymbol{a}_L$ is its output.

COMET's routing network is a second MLP with the same shape, defined by random weight matrices $\boldsymbol{V}_\ell$ (for simplicity we omit bias parameters). We sample $\boldsymbol{V}_\ell$ from the same distribution used for initializing $\boldsymbol{W}_\ell$ ($U(-N_{\ell-1}^{-1/2}, N_{\ell-1}^{-1/2})$ in our experiments). We denote this network's pre-activations and activations as $\boldsymbol{c}_\ell$ and $\boldsymbol{z}_\ell$ (analogous to $\boldsymbol{a}_\ell$ and $\boldsymbol{x}_\ell$ in the backbone network), with input $\boldsymbol{z}_0 = \boldsymbol{x}_0$. The computation of the routing network is similar to that of the backbone MLP, except that at each layer it computes a binary vector $\boldsymbol{m}_\ell$ that is then used to mask the activations in both networks. The mask is computed using a k-winner-take-all capping function C_k:

$$[C_k(\boldsymbol{v})]_i = \begin{cases} 1 & |\{j : v_j \geq v_i\}| \leq k \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

We allow a fixed proportion p_k of neurons at each layer to survive the mask, so that $k_\ell = p_\mathrm{k} N_\ell$. Then the forward pass of the routing network is defined by

$$\boldsymbol{c}_\ell = \boldsymbol{V}_\ell \boldsymbol{z}_{\ell-1} \tag{5.4}$$

$$\boldsymbol{m}_\ell = C_{k_\ell}(\boldsymbol{c}_\ell) \tag{5.5}$$

$$\boldsymbol{z}_\ell = \boldsymbol{m}_\ell \circ g(\boldsymbol{c}_\ell) \tag{5.6}$$

where $\circ$ indicates elementwise multiplication and g is the routing network's activation function (we use the identity $g(c) = c$ in the present experiments). The layerwise masks $\boldsymbol{m}_\ell$ computed by the routing network are then applied to the backbone network, so that **??** are replaced by

$$\boldsymbol{a}_\ell = \boldsymbol{W}_\ell \boldsymbol{x}_{\ell-1} \tag{5.7}$$

$$\boldsymbol{x}_\ell = \boldsymbol{m}_\ell \circ f(\boldsymbol{a}_\ell) \tag{5.8}$$

Note that the network's output $\boldsymbol{a}_L$ is computed before $\boldsymbol{m}_L$ would be applied, avoiding undesirable masking of the model's prediction.

This input-dependent masking results in a maximum number of experts that is exponential in the model size at each layer, specifically $\binom{N_\ell}{k}$. Therefore in practical settings every input will have its own expert. One consideration for this calculation might be interference among experts. Previous work has studied how multiple models can be superposed within one network [38], and [69]

show a layer with N_ℓ neurons can hold $O(e^N)$ representations that are pairwise orthogonal within a certain finite tolerance, thus minimizing interference. However, more important for the present work is that overlap between models is a desired property because it promotes generalization between similar inputs.

COMET layers differ from sparse MoE [266] layers in two major ways:

(1) **Architecturally**: Whereas a layer in a standard layered MoE architecture consists of n experts and a gating network, a COMET layer contains a random, non-trainable matrix and a k-winner-take-all cap operation. Instead of pre-defined experts, COMET layer modifies the computation of the MLP to activate only a subset of its parameters contingent on the input; this subset can be seen as an implicit expert.

(2) **In the way the information is passed**: Sparse MoE is applied in layers with a new gating network at each layer, which takes as input the backbone activation at the previous layer. Thus the gating and backbone networks at each layer take the same input. In a COMET architecture the routing network operates independently of the backbone network, so the inputs to the two are distinct (except for the first layer of the network). This ensures that a given example maps to the same implicit expert throughout both training and inference.

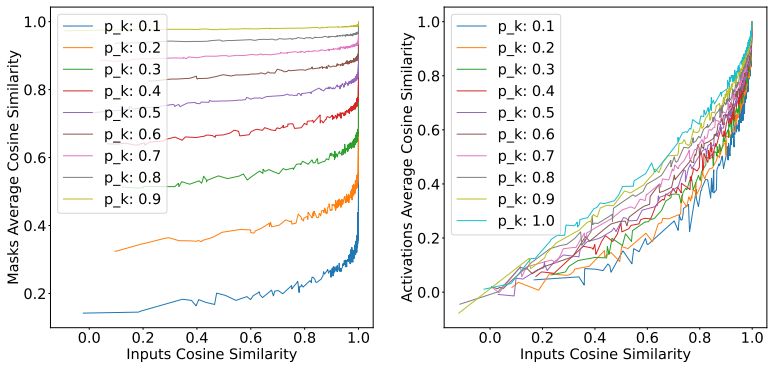
Several other important differences between existing approaches and COMET are: 1) COMET's gating network does not require training; 2) COMET does not require fixed specialization of each network module, or advanced knowledge of the combination of modules required for each task; 3) the experts in COMET overlap based on unsupervised information from input similarities; 4) COMET does not require input or task IDs to determine which mask to apply; 5) the number of possible experts in COMET is exponential in the model size.

5.3.3.1 Experiments

5.3.3.2 Synthetic Data Experiments

In this section, we describe experiments to verify key properties of a COMET network. First, we verify that the combination of the fixed routing function and cap operator maps similar inputs to similar masks and show how this sharpens the model's generalization. Second, we verify that the network makes an effective use of the available neurons.

5.3.3.3 Fixed Input-dependent Routing Network



(a) (b)

Figure 5.4: Routing properties of our gating function, which combines fixed random projections with a cap operator. (a) We compare the similarity of input pairs to the similarity of their corresponding binary masks (gates) for different sparsity levels. This plot shows that similar inputs tend to have similar masks. (b) We compare the similarity of input pairs to the similarity of their corresponding masked activation vectors in the backbone network. This plot reveals that similar inputs are mapped to similar activations, and that this relationship is sharper for sparser networks (note $p_{\mathrm{k}} = 1$ is a vanilla MLP). These properties facilitate forward knowledge transfer, even without supervision.

Our goal is to develop a fixed routing function that maps similar inputs to similar (i.e., overlapping) experts, thereby facilitating knowledge transfer between items and leading to faster learning and improved generalization.

One way to approximate generalization between individual training and test items is with the neural tangent kernel [NTK; 107]. Let $\boldsymbol{\theta} = (\boldsymbol{W}_1, \boldsymbol{b}_1, \ldots, \boldsymbol{W}_L, \boldsymbol{b}_L)$ denote the flattened concatenation of all model parameters, let $\boldsymbol{x}^{\text{train}}$ and $\boldsymbol{x}^{\text{test}}$ be arbitrary training and testing items, and let $\boldsymbol{a}_L^{\text{train}}$

and $\boldsymbol{a}_L^{\text{test}}$ be the corresponding model predictions for some fixed setting of $\boldsymbol{\theta}$. Generalization from $\boldsymbol{x}^{\text{train}}$ to $\boldsymbol{x}^{\text{test}}$ can be defined as the change in prediction $\boldsymbol{a}_L^{\text{test}}$ from including $\boldsymbol{x}^{\text{train}}$ in the training set. Formally, under a vanilla GD optimizer on loss $\mathcal{L}$, and in the limit of a small learning rate α, the contribution of $\boldsymbol{x}^{\text{train}}$ to change in $\boldsymbol{a}_L^{\text{test}}$ is

$$\frac{1}{\alpha}\Delta \boldsymbol{a}_L^{\text{test}} \underset{\alpha\to 0}{\longrightarrow} K(\boldsymbol{x}^{\text{train}}, \boldsymbol{x}^{\text{test}})\nabla_{\boldsymbol{a}_L^{\text{train}}}\mathcal{L}^{\text{train}} \tag{5.9}$$

where $K(\boldsymbol{x}^{\text{train}}, \boldsymbol{x}^{\text{test}})$ is the $N_L \times N_L$ matrix-valued NTK

$$K(\boldsymbol{x}^{\text{train}}, \boldsymbol{x}^{\text{test}}) = \frac{\partial \boldsymbol{a}_L^{\text{test}}}{\partial \boldsymbol{\theta}} \left(\frac{\partial \boldsymbol{a}_L^{\text{train}}}{\partial \boldsymbol{\theta}}\right)^\top \tag{5.10}$$

The RHS of 5.10 sums over elements of $\boldsymbol{\theta}$, and the contribution from $\boldsymbol{W}_\ell$ is

$$\sum_{ij} \frac{\partial \boldsymbol{a}_L^{\text{test}}}{\partial W_{\ell,ij}} \left(\frac{\partial \boldsymbol{a}_L^{\text{train}}}{\partial W_{\ell,ij}}\right)^\top = \sum_j a_{\ell-1,j}^{\text{test}} a_{\ell-1,j}^{\text{train}} \sum_i \frac{\partial \boldsymbol{a}_L^{\text{test}}}{\partial a_{\ell,i}^{\text{test}}} \left(\frac{\partial \boldsymbol{a}_L^{\text{train}}}{\partial a_{\ell,i}^{\text{train}}}\right)^\top \tag{5.11}$$

Thus the contribution of $\boldsymbol{W}_\ell$ to generalization from $\boldsymbol{x}^{\text{train}}$ to $\boldsymbol{x}^{\text{test}}$ is proportional to the inner product $\langle \boldsymbol{a}_{\ell-1}^{\text{train}}, \boldsymbol{a}_{\ell-1}^{\text{test}} \rangle$. This inner product will be positively related to input similarity $\langle \boldsymbol{x}^{\text{train}}, \boldsymbol{x}^{\text{test}} \rangle$ even in an unmodified MLP, but our question is how the relationship changes under COMET.

To answer this question, we conducted an experiment using 500 random pairs of input vectors. Each input had length 100 with components sampled iid from $\mathcal{N}(\mu, 25)$, with μ a random integer between 0 and 100. We calculated the cosine similarity (i.e., normalized inner product) between each pair of inputs. We then randomly initialized 10 COMET networks for every pair, each comprising a backbone network and a routing network which were both MLPs containing 10 hidden layers ($L = 11$) with 512 neurons per hidden layer. We passed both inputs through each of the 10 COMET networks, using **??** with varying degrees of sparsity p_{k}.

To analyze the behavior of our fixed gating function, we performed two complementary analyses. First, we computed the cosine similarity between the masks obtained for the two inputs in each pair, concatenated across layers as $(\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{L-1})$. Note that cosine similarity between binary vectors equals their degree of overlap, i.e. the proportion of active neurons for one input that are also active for the other. Second, we measured the cosine similarity between the two inputs'

representations in the backbone network after applying the gating function as in 5.8, again concatenating across layers as $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{L-1})$. To obtain a more robust estimate, we averaged the cosine similarities across the 10 COMET networks for each input pair, yielding the results in 5.4.

This experiment reveals that when input distributions are more similar the overlap between their binary masks increases (5.4a). This in turn strengthens the relationship between input similarity and activation similarity in the backbone network relative to the baseline MLP with $p_{\mathrm{k}} = 1$ (5.4b). Drawing on the NTK analysis above, we conclude that COMET's routing function leads the model to generalize using a narrower effective kernel. A narrower kernel should not be expected to yield universal improvement, but it should be beneficial when the base model has excess capacity for the task. The experiments in the next subsections support this prediction, in that we see an advantage for COMET particularly with larger models.

5.3.3.4 Expert Utilization

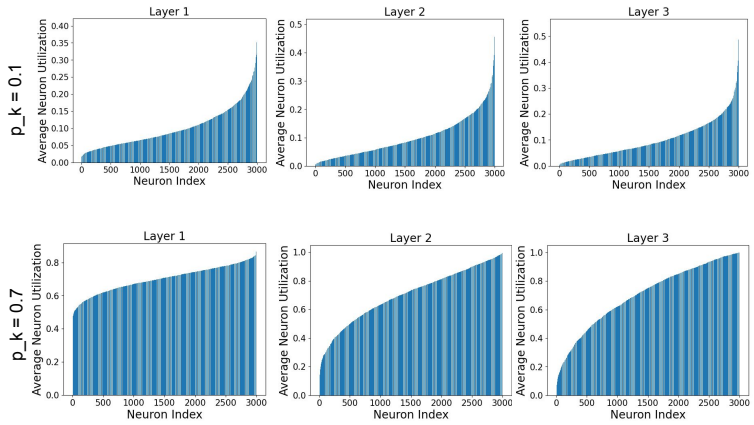


Figure 5.5: Illustration of neuron activity across COMET layers in a 4-layer MLP. We visualize the utilization of neurons in two randomly initialized networks with varying sparsity levels, using the CIFAR10 dataset. The plots show that our network effectively utilizes all its parameters, with no "dead neurons" and no signs of representation collapse, even at very high sparsity levels.

One of the challenges in training sparse architectures is representation collapse, where a small subset of experts or neurons becomes dominant, leading to under-utilization of others. This issue is particularly concerning when training gating networks, as it can result in "dead" experts or neurons

that are never activated. Given that our gating network is fixed, it is essential to investigate whether this behavior occurs, as it would be permanent.

To address this, we conducted an experiment where we generated 1000 randomly initialized 4-layer COMET MLPs with varying neuron counts per layer $N_\ell \sim U(100, 1000)$, and varying sparsity level $1 - p_\text{k} \sim U(0.05, 1)$, and passed the CIFAR10 dataset through each one. We then analyzed the utilization of each neuron, measured by how often it is activated with the given masks, and found that only $7.6e^{-4}\%$ of neurons across the ensemble had 0% utilization, and fewer than 2% of the models had any such neurons, mostly models with a high sparsity level. 5.5 presents utilization plots for two representative networks, illustrating that our fixed gating network avoids representation collapse and "dead neurons." Moreover, when we passed a different dataset (such as CIFAR100) through these models, we discovered that many previously inactive neurons were now being utilized. Thus, neurons that are infrequently utilized appear to be reserved for unseen data, highlighting the network's adaptability and capacity for generalization.

The finding is consistent with our previous analysis in 5.3.3.3, which showed that the input-dependent gating design inherently activates similar parameters for similar inputs, facilitating forward knowledge transfer. Our results suggest that our approach effectively mitigates the risk of representation collapse and promotes healthy utilization of neurons in the network, all without relying on supplementary mechanisms, such as specialized loss terms [266].

5.3.3.5 Image Classification

We extend our investigation by integrating the COMET method into a diverse range of popular architectures, including Vision Transformers (ViTs), MLP-Mixers, and standard MLPs.

5.3.3.6 Standard MLP – CIFAR10

We apply the COMET method to a standard MLP with 4 layers, varying the number of neurons in each layer and the sparsity levels. To evaluate its performance, we compare it to 10 related methods:

Standard Model: A standard MLP model with the same number of neurons and no sparsity.

Smaller Model: A smaller model with a reduced number of neurons, specifically $p_{\mathrm{k}} N_\ell$ where N_ℓ is the width of the standard model.

Dropout Model: A standard model with a dropout rate equal to $1 - p_{\mathrm{k}}$.

Topk Model: An MLP with a trainable routing function. The cap operation is applied directly to the backbone network by replacing 5.5 with $\boldsymbol{m}_\ell = C_{k_\ell}(\boldsymbol{a}_\ell)$, so that the routing function selects the highest k values and masks the remaining ones.

MoE Trainable: A MoE model with $\lfloor 1/p_{\mathrm{k}} \rfloor$ experts, each having $p_{\mathrm{k}} N_\ell$ neurons in each layer. The routing network is a trainable MLP with one hidden layer and a sparse $\lfloor 1/p_{\mathrm{k}} \rfloor$-dimensional output.

MoE Non-trainable: Same as MoE Trainable, with a fixed routing function.

Layer-wise Routing: An MLP where each backbone hidden layer representation is projected using a fixed random matrix, which is then used to develop the binary mask for the next layer. This is done by replacing 5.4 with $\boldsymbol{c}_\ell = \boldsymbol{V}_\ell \boldsymbol{x}_{\ell-1}$.

Bernoulli Masking: An MLP where each training example is associated with a fixed binary mask drawn from a Bernoulli distribution, with probability equal to p_{k}. Thus the relationship between inputs and their masks is arbitrary, rather than being mediated by the routing network in COMET.

Example-tied Dropout: Example-tied dropout [170], where each example in the training data is associated with a fixed binary mask drawn from a Bernoulli distribution, with probability equal to p_{k}, and a fixed number of “generalization neurons" are active for all examples.

Standard model L1: A standard MLP model, but using L1 regularization to induce sparsity.

We evaluate these models on the CIFAR10 dataset [137], with results shown in 5.6. Overall, the optimal model architecture depends on the capacity of the network. When the number of neurons is limited and the network has a low capacity to learn the task (i.e., low p_{k}), the standard model that utilizes all neurons outperforms most models. However, as network capacity increases with more neurons, the COMET model emerges as the top performer. This suggests that the benefits of selective neuron activation become more pronounced as capacity increases. Notice in the high-capacity regime the Smaller Model matches the Standard Model, indicating that simply adding

more neurons does not improve performance while adding neurons subject to COMET's structured sparsity does.

5.3.3.7 Contemporary Architectures

We further extend the COMET method to contemporary architectures in the Vision domain, including ViT [62] and MLP-Mixer [289]. To do this, we apply the COMET random projection followed by the cap operation in the MLP layers of each with $p_{\text{k}} = 0.5$. We evaluate the performance of these models on four widely-used image classification datasets: SVHN [201], CIFAR10 [137], CIFAR100 [137], and Tiny ImageNet [141]. Our results can be seen in 5.7, 5.8.

A similar trend emerges in these architectures: as network capacity increases, the optimal model architecture shifts. In smaller networks, where the number of neurons in the MLP layer is limited, the standard model performs roughly similarly to the COMET model. However, even in these networks, incorporating COMET layers yields notable performance improvements. As we scale up the network by adding more neurons, COMET displays superior performance across all five model architectures and four datasets. It achieves faster convergence and significantly higher accuracy, with gains of up to 9% in ViT Large on CIFAR100. Moreover, we observe that the performance gap between the COMET-based models and their standard counterparts widens as the model size increases, with larger models exhibiting both better performance and faster learning rates. This reinforces our finding that selective neuron activation becomes increasingly beneficial as network capacity grows.

5.3.3.8 Language Modeling and Regression

We apply COMET to language modeling on Wikitext [182] and CodeParrot [291] with varying GPT model sizes, with results in 5.9 and 5.10. We again observe that as network capacity increases, the COMET model outperforms the standard model, with larger models exhibiting not only a greater performance difference but also faster learning rates, highlighting the benefits of selective neuron activation in language modeling tasks.

To further validate our results, we also evaluated COMET on the SARCOS regression dataset and show that our conclusions generalize to this setting as well.

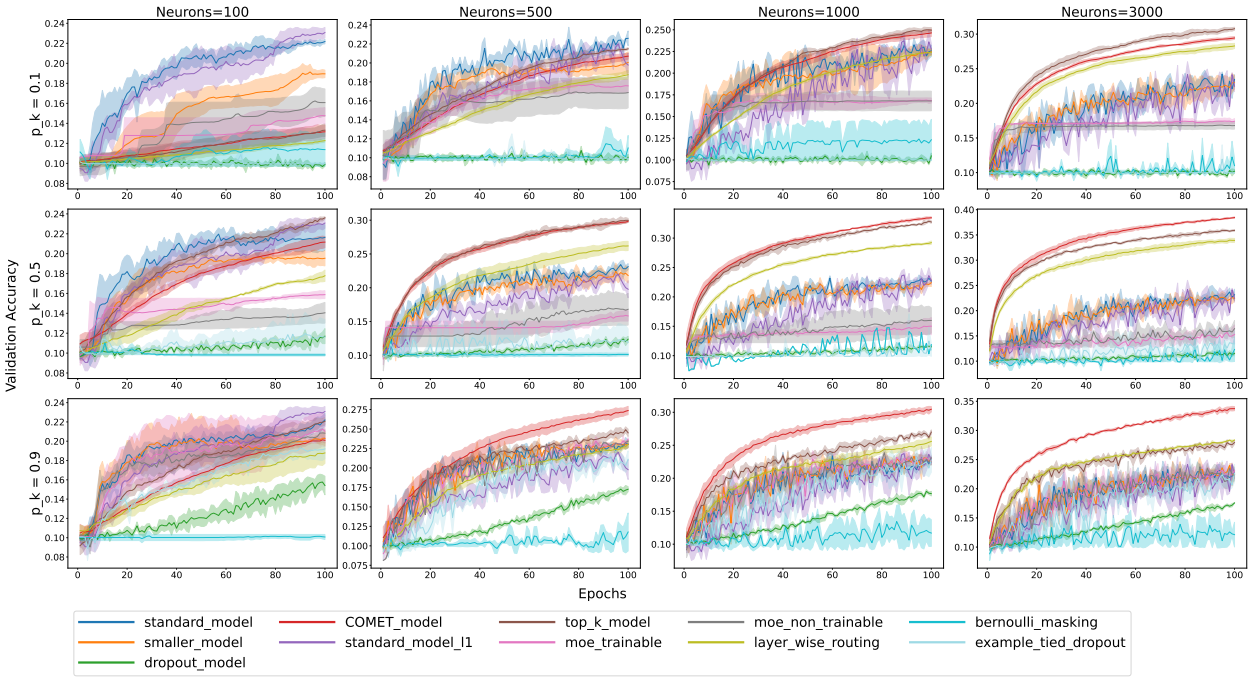


Figure 5.6: Illustration of 4-layer MLP networks trained on CIFAR10, showcasing the impact of varying network capacity and sparsity levels. As we increase the number of neurons and decrease sparsity (moving from top left to bottom right), we observe a shift in the best-performing model. Initially, the standard model outperforms the COMET model when network capacity is low. However, as network capacity grows, the COMET model emerges as the top performer.

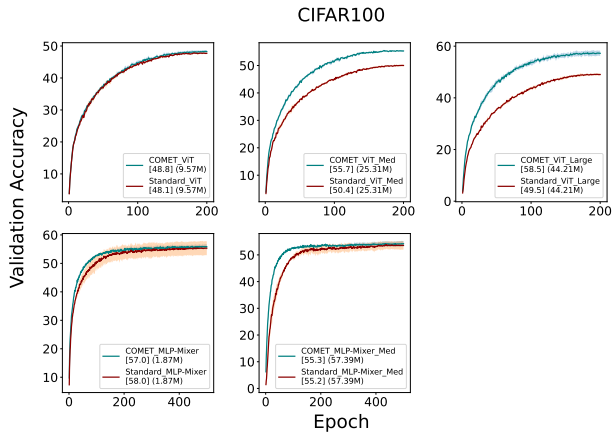


Figure 5.7: ViTs and MLP-Mixers on CIFAR100. [Highest accuracy] (# trainable param.)

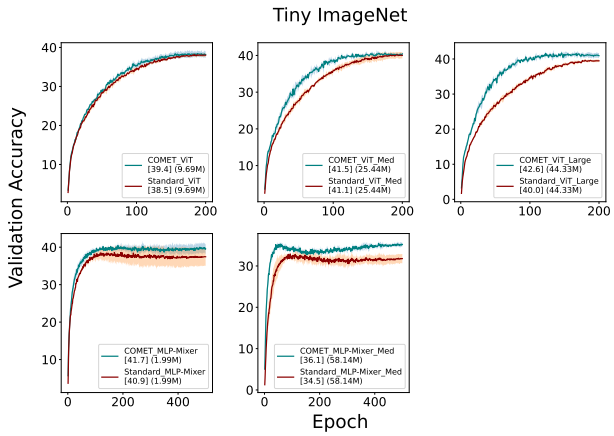


Figure 5.8: ViTs and MLP-Mixers on Tiny ImageNet. [Highest accuracy] (# trainable param.)

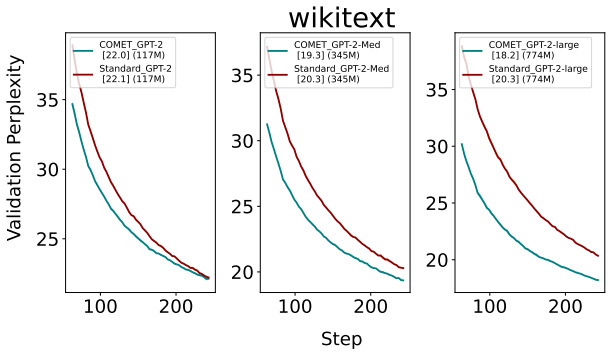


Figure 5.9: GPTs trained on WikiText. [Lowest perplexity] (# trainable param.)

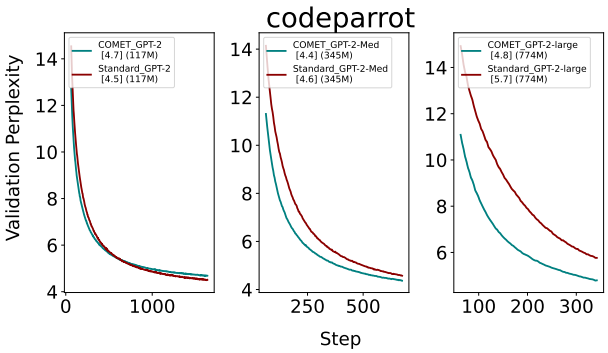


Figure 5.10: GPTs trained on CodeParrot. [Lowest perplexity] (# trainable param.)

Chapter 6

Future Work

Building upon the findings from this research, several directions for future work emerge. The studies have highlighted key challenges in fairness, knowledge assessment, contextual understanding, citation generation, and modular architectures, all of which present opportunities for further investigation and refinement.

One critical avenue for future research is ensuring fairness and mitigating bias in biomedical QA systems. The findings demonstrate that irrelevant demographic information affects model predictions, underscoring the need for techniques that improve robustness and fairness. Future work could explore methods such as adversarial training, domain adaptation, or bias correction strategies to ensure that QA systems provide equitable responses regardless of demographic factors. Additionally, expanding these analyses to larger datasets and real-world clinical settings could provide further insights into model biases and mitigation strategies.

Another important direction involves improving knowledge probing techniques. The study on template-based vs. template-free probing methods revealed substantial differences in assessing LMs' knowledge. Future research should focus on developing hybrid probing techniques that combine the benefits of both approaches while minimizing their respective limitations. Additionally, extending template-free probing to other domains, including multilingual and low-resource settings, could enhance understanding of knowledge representation across diverse linguistic and cultural contexts.

In the realm of reading comprehension, the work highlights the impact of input ordering and emphasis on model performance. Future efforts should investigate whether these effects generalize

across broader model architectures and additional tasks, such as reasoning-based QA or multi-hop inference. Moreover, developing dynamic input manipulation techniques that adapt to different question types and model architectures could further optimize performance.

The evaluation of context-based QA desiderata provides an opportunity for refining evaluation methodologies. Future work could extend this research by incorporating human-in-the-loop evaluations, investigating interactive QA models, and designing systems that dynamically adjust to user preferences. Additionally, exploring how desiderata interact in multi-modal settings, where textual, visual, and structured data are combined, could further enhance QA system capabilities.

The need for citation-aware language models remains an open challenge. While this research outlines the importance of models that can cite sources to enhance trustworthiness, future research should focus on developing scalable RAG techniques that align model outputs with verifiable sources. This includes integrating retrieval mechanisms directly into pretraining, designing evaluation metrics that measure citation quality, and addressing challenges in hallucination and source attribution.

In ambiguous QA settings, future work should refine the proposed framework for generating and evaluating cited responses. Exploring how citation mechanisms can adapt to different domains, such as legal or biomedical question answering, could improve model transparency and user trust. Furthermore, investigating how citation-driven QA can be integrated with existing fact-checking pipelines could lead to more robust and reliable information retrieval systems.

Finally, in the area of modular and sparse neural architectures, the work on COMET opens multiple avenues for improvement. Future research could investigate adaptive methods for expert selection, explore reinforcement learning-based gating mechanisms, and analyze the interplay between modularity and generalization across broader AI applications. Additionally, extending COMET's principles to lifelong learning and continual adaptation scenarios could enable more flexible and scalable architectures.

Overall, the findings from this dissertation provide a foundation for multiple future research directions aimed at enhancing the fairness, performance, and efficiency of LLMs.

Bibliography

[1] Imen Akermi, Johannes Heinecke, and Frédéric Herledan. Transformer based natural language generation for question-answering. In Proceedings of the 13th International Conference on Natural Language Generation, pages 349–359, Dublin, Ireland, December 2020. Association for Computational Linguistics.

[2] Dmitriy Alexandrov, Anastasiia Zakharova, and Nikolay Butakov. Does noise really matter? investigation into the influence of noisy labels on bert-based question answering system. In 2023 IEEE 17th International Conference on Semantic Computing (ICSC), pages 33–40, 2023.

[3] Zafar Ali, Guilin Qi, Khan Muhammad, Siddhartha Bhattacharyya, Irfan Ullah, and Waheed Abro. Citation recommendation employing heterogeneous bibliographic network embedding. Neural Comput. Appl., 34(13):10229–10242, jul 2022.

[4] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning, 2019.

[5] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of open language models, 2023.

[6] Sultan Alrowili and Vijay Shanker. BioM-transformers: Building large biomedical language models with BERT, ALBERT and ELECTRA. In Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi Tsujii, editors, Proceedings of the 20th Workshop on Biomedical Language Processing, pages 221–227, Online, June 2021. Association for Computational Linguistics.

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

[8] Josh Andle, Ali Payani, and Salimeh Yasaei-Sekeh. Investigating the impact of weight sharing decisions on knowledge transfer in continual learning, 2023.

[9] Anthropic. Model Card and Evaluations for Claude Models Anthropic.

[10] Stéphane Aroca-Ouellette, Cory Paik, Alessandro Roncone, and Katharina Kann. Prost: Physical reasoning of objects through space and time, 2021.

[11] Simran Arora, Sen Wu, Enci Liu, and Christopher Re. Metadata shaping: A simple approach for knowledge-enhanced language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Findings of the Association for Computational Linguistics: ACL 2022, pages 1733–1745, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[12] Arian Askari, Amin Abolghasemi, Gabriella Pasi, Wessel Kraaij, and Suzan Verberne. Injecting the bm25 score as text improves bert-based re-rankers, 2023.

[13] Jimmy Ba and Brendan Frey. Adaptive dropout for training deep neural networks. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 26. Curran Associates, Inc., 2013.

[14] Jinheon Baek, Alham Aji, and Amir Saffari. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In Estevam Hruschka, Tom Mitchell, Sajjadur Rahman, Dunja Mladenić, and Marko Grobelnik, editors, Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023), pages 70–98, Toronto, ON, Canada, July 2023. Association for Computational Linguistics.

[15] Hossein Bahak, Farzaneh Taheri, Zahra Zojaji, and Arefeh Kazemi. Evaluating chatgpt as a question answering system: A comprehensive analysis and comparison with existing models, 2023.

[16] Anna Bair, Hongxu Yin, Maying Shen, Pavlo Molchanov, and Jose Alvarez. Adaptive sharpness-aware pruning for robust sparse networks, 2024.

[17] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension dataset, 2018.

[18] Rongzhou Bao, Jiayi Wang, and Hai Zhao. Defending pre-trained language models from adversarial word substitution without performance sacrifice. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3248–3258, Online, August 2021. Association for Computational Linguistics.

[19] Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. Beat the AI: Investigating adversarial human annotation for reading comprehension. Transactions of the Association for Computational Linguistics, 8:662–678, 2020.

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

[21] Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models, 2016.

[22] Yoshua Bengio. Deep learning of representations: Looking forward, 2013.

[23] Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (technology) is power: A critical survey of "bias" in NLP. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for

Computational Linguistics, pages 5454–5476, Online, July 2020. Association for Computational Linguistics.

[24] Bernd Bohnet, Vinh Q. Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, Tom Kwiatkowski, Ji Ma, Jianmo Ni, Lierni Sestorain Saralegui, Tal Schuster, William W. Cohen, Michael Collins, Dipanjan Das, Donald Metzler, Slav Petrov, and Kellie Webster. Attributed question answering: Evaluation and modeling for attributed large language models, 2023.

[25] Marissa Borgese, Cara Joyce, Emily E Anderson, Matthew M Churpek, and Majid Afshar. Bias assessment and correction in machine learning algorithms: A use-case in a natural language processing algorithm to identify hospitalized patients with unhealthy alcohol use. AMIA Annu. Symp. Proc., 2021:247–254, 2021.

[26] Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. Inducing relational knowledge from bert, 2019.

[27] Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. Large language models in machine translation. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 858–867, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

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

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

[30] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[31] Nina Dekoninck Bruhin and Bryn Davies. Bioinspired random projections for robust, sparse classification, 2022.

[32] Meng Cao, Yue Dong, and Jackie Cheung. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3340–3354, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[33] Yu Cao, Dianqi Li, Meng Fang, Tianyi Zhou, Jun Gao, Yibing Zhan, and Dacheng Tao. Tasa: Deceiving question answering models by twin answer sentences attack, 2022.

[34] Hung-Ting Chen, Michael Zhang, and Eunsol Choi. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 2292–2307, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[35] Tianlong Chen, Zhenyu Zhang, Ajay Jaiswal, Shiwei Liu, and Zhangyang Wang. Sparse moe as the new dropout: Scaling dense and self-slimmable transformers, 2023.

[36] Zhourong Chen, Yang Li, Samy Bengio, and Si Si. You look twice: Gaternet for dynamic filter selection in cnns, 2019.

[37] Daixuan Cheng, Shaohan Huang, and Furu Wei. Adapting large language models via reading comprehension, 2023.

[38] Brian Cheung, Alex Terekhov, Yubei Chen, Pulkit Agrawal, and Bruno Olshausen. Superposition of many models into one, 2019.

[39] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. QuAC: Question answering in context. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2174–2184, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[40] Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. Coarse-to-fine question answering for long documents. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 209–220, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[41] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee,

Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[42] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[43] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[44] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

[45] Kenneth Church, Valia Kordoni, Gary Marcus, Ernest Davis, Yanjun Ma, and Zeyu Chen. A gentle introduction to deep nets and opportunities for the future. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, pages 1–6, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[46] Christopher Clark and Matt Gardner. Simple and effective multi-paragraph reading comprehension. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 845–855, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[47] Roi Cohen, Mor Geva, Jonathan Berant, and Amir Globerson. Crawling the internal knowledge-base of language models, 2023.

[48] Jeremy R. Cole, Michael J. Q. Zhang, Daniel Gillick, Julian Martin Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions, 2023.

[49] Robert Dale. Gpt-3: What's it good for? Natural Language Engineering, 27(1):113–118, 2021.

[50] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. A survey of the state of explainable AI for natural language processing. In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 447–459, Suzhou, China, December 2020. Association for Computational Linguistics.

[51] Maxime De Bruyn, Ehsan Lotfi, Jeska Buhmann, and Walter Daelemans. Is it smaller than a tennis ball? language models play the game of twenty questions. In Jasmijn Bastings, Yonatan Belinkov, Yanai Elazar, Dieuwke Hupkes, Naomi Saphra, and Sarah Wiegreffe, editors, Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 80–90, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

[52] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models, 2021.

[53] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models, 2021.

[54] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models, 2021.

[55] Yang Deng, Wenxuan Zhang, Yifei Yuan, and Wai Lam. Knowledge-enhanced mixed-initiative dialogue system for emotional support conversations. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4079–4095, Toronto, Canada, July 2023. Association for Computational Linguistics.

[56] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[57] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[58] Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. Time-aware language models as temporal knowledge bases. Transactions of the Association for Computational Linguistics, 10:257–273, 2022.

[59] Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. Calibrating factual knowledge in pretrained language models, 2022.

[60] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning, 2024.

[61] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. A survey on in-context learning, 2023.

[62] 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, 2021.

[63] Haowei Du, Quzhe Huang, Chen Zhang, and Dongyan Zhao. Knowledge-enhanced iterative instruction generation and reasoning for knowledge base question answering, 2022.

[64] Daria Dzendzik, Jennifer Foster, and Carl Vogel. English machine reading comprehension datasets: A survey. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 8784–8804, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[65] Nouha Dziri, Andrea Madotto, Osmar Zaïane, and Avishek Joey Bose. Neural path hunter: Reducing hallucination in dialogue systems via path grounding. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2197–2214, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[66] Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. On the origin of hallucinations in conversational models: Is it the datasets or the models? In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5271–5285, Seattle, United States, July 2022. Association for Computational Linguistics.

[67] Jacob Eisenstein, Daniel Andor, Bernd Bohnet, Michael Collins, and David Mimno. Honest students from untrusted teachers: Learning an interpretable question-answering pipeline from a pretrained language model, 2022.

[68] Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models, 2021.

[69] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. arXiv preprint arXiv:2209.10652, 2022.

[70] Jeffrey L. Elman. Finding structure in time. Cognitive Science, 14:179–211, 1990.

[71] Li Fang, Qingyu Chen, Chih-Hsuan Wei, Zhiyong Lu, and Kai Wang. Bioformer: an efficient transformer language model for biomedical text mining, 2023.

[72] Tianqing Fang, Zhaowei Wang, Wenxuan Zhou, Hongming Zhang, Yangqiu Song, and Muhao Chen. Getting sick after seeing a doctor? diagnosing and mitigating knowledge conflicts in event temporal reasoning, 2023.

[73] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2022.

[74] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks, 2017.

[75] Besnik Fetahu, Katja Markert, Wolfgang Nejdl, and Avishek Anand. Finding news citations for wikipedia. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. ACM, oct 2016.

[76] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks, 2019.

[77] Robert M. French. Using semi-distributed representations to overcome catastrophic forgetting in connectionist networks, 1993.

[78] Wee Chung Gan and Hwee Tou Ng. Improving the robustness of question answering systems to question paraphrasing. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6065–6075, Florence, Italy, July 2019. Association for Computational Linguistics.

[79] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020.

[80] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 16477–16508, Toronto, Canada, July 2023. Association for Computational Linguistics.

[81] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations, 2023.

[82] Yifan Gao, Henghui Zhu, Patrick Ng, Cicero Nogueira dos Santos, Zhiguo Wang, Feng Nan, Dejiao Zhang, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. Answering ambiguous questions through generative evidence fusion and round-trip prediction. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3263–3276, Online, August 2021. Association for Computational Linguistics.

[83] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[84] Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1307–1323, Online, November 2020. Association for Computational Linguistics.

[85] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories, 2020.

[86] L. Gierth and R. Bromme. Beware of vested interests: Epistemic vigilance improves reasoning about scientific evidence (for some people). PLoS One, 15(4):e0231387, 2020.

[87] Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. Improving alignment of dialogue agents via targeted human judgements, 2022.

[88] Nianlong Gu and Richard H. R. Hahnloser. Controllable citation text generation, 2022.

[89] 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, 2020.

[90] Reto Gubelmann and Siegfried Handschuh. Context matters: A pragmatic study of PLMs' negation understanding. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4602–4621, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[91] Nuno M. Guerreiro, Duarte Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. Hallucinations in large multilingual translation models, 2023.

[92] Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. DEMix layers: Disentangling domains for modular language modeling. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5557–5576, Seattle, United States, July 2022. Association for Computational Linguistics.

[93] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Proceedings of ACL, 2020.

[94] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[95] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, 2016.

[96] Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. A survey on recent approaches for natural language processing in low-resource scenarios. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2545–2568, Online, June 2021. Association for Computational Linguistics.

[97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[98] Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks, 2021.

[99] Giwon Hong, Jeonghwan Kim, Junmo Kang, Sung-Hyon Myaeng, and Joyce Jiyoung Whang. Discern and answer: Mitigating the impact of misinformation in retrieval-augmented models with discriminators, 2023.

[100] Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. True: Re-evaluating factual consistency evaluation, 2022.

[101] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[102] Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. A survey of knowledge enhanced pre-trained language models, 2023.

[103] Hao Huang, Xiubo Geng, Guodong Long, and Daxin Jiang. Understand before answer: Improve temporal reading comprehension via precise question understanding. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 375–384, Seattle, United States, July 2022. Association for Computational Linguistics.

[104] Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning, 2019.

[105] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. Neural Computation, 3(1):79–87, 1991.

[106] Tom Jacobs and Rebekka Burkholz. Mask in the mirror: Implicit sparsification, 2024.

[107] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks, 2020.

[108] Katharina Jeblick, Balthasar Schachtner, Jakob Dexl, Andreas Mittermeier, Anna Theresa Stüber, Johanna Topalis, Tobias Weber, Philipp Wesp, Bastian Sabel, Jens Ricke, and Michael Ingrisch. Chatgpt makes medicine easy to swallow: An exploratory case study on simplified radiology reports, 2022.

[109] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[110] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

[111] Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 7969–7992, Singapore, December 2023. Association for Computational Linguistics.

[112] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know?, 2019.

[113] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? Transactions of the Association for Computational Linguistics, 8:423–438, 2020.

[114] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know?, 2020.

[115] 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, 11(14), 2021.

[116] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the em algorithm. In Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), volume 2, pages 1339–1344 vol.2, 1993.

[117] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[118] Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. Maieutic prompting: Logically consistent reasoning with recursive explanations, 2022.

[119] Dan Jurafsky and James Martin. Speech and Language Processing. Pearson, Upper Saddle River, NJ, 2 edition, May 2008.

[120] Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. Evaluating open-domain question answering in the era of large language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5591–5606, Toronto, Canada, July 2023. Association for Computational Linguistics.

[121] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge, 2023.

[122] Haeyong Kang, Jaehong Yoon, Sung Ju Hwang, and Chang D. Yoo. Continual learning: Forget-free winning subnetworks for video representations, 2024.

[123] Georgi Karadzhov, Preslav Nakov, Lluis Marquez, Alberto Barron-Cedeno, and Ivan Koychev. Fully automated fact checking using external sources, 2017.

[124] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6769–6781, Online, November 2020. Association for Computational Linguistics.

[125] Nora Kassner, Philipp Dufter, and Hinrich Schütze. Multilingual LAMA: Investigating knowledge in multilingual pretrained language models. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 3250–3258, Online, April 2021. Association for Computational Linguistics.

[126] Nora Kassner and Hinrich Schütze. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7811–7818, Online, July 2020. Association for Computational Linguistics.

[127] Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 8849–8861, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[128] Divyansh Kaushik and Zachary C. Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks, 2018.

[129] Divyansh Kaushik and Zachary C. Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 5010–5015, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[130] Mehran Kazemi, Quan Yuan, Deepti Bhatia, Najoung Kim, Xin Xu, Vaiva Imbrasaite, and Deepak Ramachandran. Boardgameqa: A dataset for natural language reasoning with contradictory information, 2023.

[131] Rohit Keshari, Richa Singh, and Mayank Vatsa. Guided dropout, 2018.

[132] Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. More bang for your buck: Natural perturbation for robust question answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 163–170, Online, November 2020. Association for Computational Linguistics.

[133] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[134] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023.

[135] Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, Roger A Sayle, Riza Theresa Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rocktäschel, Sérgio Matos, David Campos, Buzhou Tang, Hua Xu, Tsendsuren Munkhdalai, Keun Ho Ryu, SV Ramanan, Senthil Nathan, Slavko Žitnik, Marko Bajec, Lutz Weber, Matthias Irmer, Saber A Akhondi, Jan A Kors, Shuo Xu, Xin An, Utpal Kumar Sikdar, Asif Ekbal, Masaharu Yoshioka, Thaer M Dieb, Miji Choi, Karin Verspoor, Madian Khabsa, C Lee Giles, Hongfang Liu, Komandur Elayavilli Ravikumar, Andre Lamurias, Francisco M Couto, Hong-Jie Dai, Richard Tzong-Han Tsai, Caglar Ata, Tolga Can, Anabel Usié, Rui Alves, Isabel Segura-Bedmar, Paloma Martínez, Julen Oyarzabal, and Alfonso Valencia. The CHEMDNER corpus of chemicals and drugs and its annotation principles. Journal of Cheminformatics, 7(S1), January 2015.

[136] F. V. Krasnova, I. S. Smaznevicha, and E. N. Baskakova. Text sampling strategies for predicting missing bibliographic links, 2023.

[137] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical Report, University of Toronto, 2009.

[138] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7:452–466, 2019.

[139] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations, 2017.

[140] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2020.

[141] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015.

[142] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In D. Touretzky, editor, Advances in Neural Information Processing Systems, volume 2. Morgan-Kaufmann, 1989.

[143] Nayeon Lee, Belinda Z. Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khabsa. Language models as fact checkers?, 2020.

[144] Sharon Levy, Tahilin Sanchez Karver, William D. Adler, Michelle R. Kaufman, and Mark Dredze. Evaluating biases in context-dependent health questions, 2024.

[145] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[146] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[147] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[148] Bo Li, Yifei Shen, Jingkang Yang, Yezhen Wang, Jiawei Ren, Tong Che, Jun Zhang, and Ziwei Liu. Sparse mixture-of-experts are domain generalizable learners, 2023.

[149] Dacheng Li, Rulin Shao*, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, , and Hao Zhang. How long can open-source llms truly promise on context length?, June 2023.

[150] Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. Large language models with controllable working memory, 2022.

[151] Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 6449–6464, Singapore, December 2023. Association for Computational Linguistics.

[152] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-ATTACK: Adversarial attack against BERT using BERT. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6193–6202, Online, November 2020. Association for Computational Linguistics.

[153] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang,

Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models, 2022.

[154] Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. Learning language specific sub-network for multilingual machine translation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 293–305, Online, August 2021. Association for Computational Linguistics.

[155] Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. Self-alignment pretraining for biomedical entity representations, 2020.

[156] Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. Evaluating the logical reasoning ability of chatgpt and gpt-4, 2023.

[157] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023.

[158] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021.

[159] 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, abs/1907.11692, 2019.

[160] 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, 2019.

[161] Yujian Liu, Jiabao Ji, Tong Yu, Ryan Rossi, Sungchul Kim, Handong Zhao, Ritwik Sinha, Yang Zhang, and Shiyu Chang. Augment before you try: Knowledge-enhanced table question answering via table expansion, 2024.

[162] Cécile Logé, Emily Ross, David Yaw Amoah Dadey, Saahil Jain, Adriel Saporta, Andrew Y. Ng, and Pranav Rajpurkar. Q-pain: A question answering dataset to measure social bias in pain management, 2021.

[163] Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7052–7063, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[164] Man Luo, Kazuma Hashimoto, Semih Yavuz, Zhiwei Liu, Chitta Baral, and Yingbo Zhou. Choose your QA model wisely: A systematic study of generative and extractive readers for question answering. In Rajarshi Das, Patrick Lewis, Sewon Min, June Thai, and Manzil Zaheer, editors, Proceedings of the 1st Workshop on Semiparametric Methods in NLP: Decoupling Logic from Knowledge, pages 7–22, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics.

[165] Kelvin Luu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A Smith. Citation text generation. arXiv preprint arXiv:2002.00317, 2020.

[166] Kai Lv, Yuqing Yang, Tengxiao Liu, Qinghui Gao, Qipeng Guo, and Xipeng Qiu. Full parameter fine-tuning for large language models with limited resources, 2024.

[167] Maria Lymperaiou and Giorgos Stamou. A survey on knowledge-enhanced multimodal learning, 2024.

[168] Manuel Mager, Ximena Gutierrez-Vasques, Gerardo Sierra, and Ivan Meza-Ruiz. Challenges of language technologies for the indigenous languages of the Americas. In Proceedings of the 27th International Conference on Computational Linguistics, pages 55–69, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[169] Adyasha Maharana and Mohit Bansal. Adversarial augmentation policy search for domain and cross-lingual generalization in reading comprehension. In Trevor Cohn, Yulan He, and Yang Liu, editors, Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3723–3738, Online, November 2020. Association for Computational Linguistics.

[170] Pratyush Maini, Michael C. Mozer, Hanie Sedghi, Zachary C. Lipton, J. Zico Kolter, and Chiyuan Zhang. Can neural network memorization be localized?, 2023.

[171] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics.

[172] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights, 2018.

[173] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning, 2018.

[174] Gary Ernest Davis Marcus. Rebooting AI: Building Artificial Intelligence we can Trust. 2019.

[175] Alejandro Martín, Javier Huertas-Tato, Álvaro Huertas-García, Guillermo Villar-Rodríguez, and David Camacho. Facter-check: Semi-automated fact-checking through semantic similarity and natural language inference, 2021.

[176] Nicolas Y. Masse, Gregory D. Grant, and David J. Freedman. Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. Proceedings of the National Academy of Sciences, 115(44), October 2018.

[177] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1906–1919, Online, July 2020. Association for Computational Linguistics.

[178] Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics.

[179] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2022.

[180] Zaiqiao Meng, Fangyu Liu, Ehsan Shareghi, Yixuan Su, Charlotte Collins, and Nigel Collier. Rewire-then-probe: A contrastive recipe for probing biomedical knowledge of pre-trained language models, 2021.

[181] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. Teaching language models to support answers with verified quotes, 2022.

[182] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

[183] Paul Michel, Xian Li, Graham Neubig, and Juan Pino. On evaluation of adversarial perturbations for sequence-to-sequence models. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 3103–3114, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[184] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering, 2018.

[185] Tsvetomila Mihaylova, Preslav Nakov, Lluis Marquez, Alberto Barron-Cedeno, Mitra Mohtarami, Georgi Karadzhov, and James Glass. Fact checking in community forums, 2018.

[186] Sewon Min, Kenton Lee, Ming-Wei Chang, Kristina Toutanova, and Hannaneh Hajishirzi. Joint passage ranking for diverse multi-answer retrieval. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6997–7008, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[187] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In EMNLP, 2020.

[188] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5783–5797, Online, November 2020. Association for Computational Linguistics.

[189] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale, 2021.

[190] Sarthak Mittal, Yoshua Bengio, and Guillaume Lajoie. Is a modular architecture enough?, 2022.

[191] Azade Mohammadi, Reza Ramezani, and Ahmad Baraani. A comprehensive survey on multihop machine reading comprehension approaches, 2022.

[192] Sungrim (Riea) Moon and Jungwei Fan. How you ask matters: The effect of paraphrastic questions to BERT performance on a clinical SQuAD dataset. In Proceedings of the 3rd Clinical Natural Language Processing Workshop, pages 111–116, Online, November 2020. Association for Computational Linguistics.

[193] Hesham Mostafa and Xin Wang. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization, 2019.

[194] MPT. Introducing mpt-7b: A new standard for open-source, commercially usable llms, May 2023.

[195] Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. Did the model understand the question?, 2018.

[196] Mohammed Muqeeth, Haokun Liu, and Colin Raffel. Models with conditional computation learn suboptimal solutions, 2022.

[197] Moin Nadeem, Anna Bethke, and Siva Reddy. StereoSet: Measuring stereotypical bias in pretrained language models. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 5356–5371, Online, August 2021. Association for Computational Linguistics.

[198] Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, Dragomir Radev, and Dragomir Radev. FeTaQA: Free-form table question answering. Transactions of the Association for Computational Linguistics, 10:35–49, 2022.

[199] Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. Disentqa: Disentangling parametric and contextual knowledge with counterfactual question answering, 2022.

[200] Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. DisentQA: Disentangling parametric and contextual knowledge with counterfactual question answering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10056–10070, Toronto, Canada, July 2023. Association for Computational Linguistics.

[201] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.

[202] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In Proceedings of the twenty-first international conference on Machine learning, page 78, 2004.

[203] Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2335–2345, Florence, Italy, July 2019. Association for Computational Linguistics.

[204] Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. Capabilities of gpt-4 on medical challenge problems, 2023.

[205] Yasumasa Onoe, Michael Zhang, Eunsol Choi, and Greg Durrett. Entity cloze by date: What LMs know about unseen entities. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 693–702, Seattle, United States, July 2022. Association for Computational Linguistics.

[206] OpenAI. Chatgpt: Optimizing language models for dialogue, Jan 2023.

[207] OpenAI. Gpt-4 technical report, 2023.

[208] Kalyani Pakhale. Comprehensive overview of named entity recognition: Models, domain-specific applications and challenges, 2023.

[209] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa : A large-scale multi-subject multi-choice dataset for medical domain question answering, 2022.

[210] Liangming Pan, Wenhu Chen, Min-Yen Kan, and William Yang Wang. Contraqa: Question answering under contradicting contexts, 2021.

[211] Liangming Pan, Xinyuan Lu, Min-Yen Kan, and Preslav Nakov. QACheck: A demonstration system for question-guided multi-hop fact-checking. In Yansong Feng and Els Lefever, editors, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 264–273, Singapore, December 2023. Association for Computational Linguistics.

[212] Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Yang Wang. On the risk of misinformation pollution with large language models, 2023.

[213] Dimitris Pappas, Ion Androutsopoulos, and Haris Papageorgiou. BioRead: A new dataset for biomedical reading comprehension. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[214] Dimitris Pappas, Petros Stavropoulos, Ion Androutsopoulos, and Ryan McDonald. BioMRC: A dataset for biomedical machine reading comprehension. In Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi Tsujii, editors, Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing, pages 140–149, Online, July 2020. Association for Computational Linguistics.

[215] Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities, 2024.

[216] Mansheej Paul, Feng Chen, Brett W. Larsen, Jonathan Frankle, Surya Ganguli, and Gintare Karolina Dziugaite. Unmasking the lottery ticket hypothesis: What's encoded in a winning ticket's mask? In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023.

[217] Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. ArXiv, abs/1906.05474, 2019.

[218] Lorenzo Pes, Rick Luiken, Federico Corradi, and Charlotte Frenkel. Active dendrites enable efficient continual learning in time-to-first-spike neural networks, 2024.

[219] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases?, 2019.

[220] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases?, 2019.

[221] Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Maria Ponti. Modular deep learning, 2024.

[222] Rebecca Qian, Candace Ross, Jude Fernandes, Eric Michael Smith, Douwe Kiela, and Adina Williams. Perturbation augmentation for fairer NLP. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 9496–9521, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[223] Zackary Rackauckas. Rag-fusion: A new take on retrieval augmented generation. International Journal on Natural Language Computing, 13(1):37–47, February 2024.

[224] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[225] Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.

[226] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[227] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

[228] 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, 2019.

[229] 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, 2023.

[230] Nasim Rahaman, Muhammad Waleed Gondal, Shruti Joshi, Peter Gehler, Yoshua Bengio, Francesco Locatello, and Bernhard Schölkopf. Dynamic inference with neural interpreters, 2021.

[231] Hadsell R;Rao D;Rusu AA;Pascanu Raia. Embracing change: Continual learning in deep neural networks, 2020.

[232] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[233] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras, editors, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[234] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.

[235] Carl Rasmussen and Zoubin Ghahramani. Infinite mixtures of gaussian process experts. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, Advances in Neural Information Processing Systems, volume 14. MIT Press, 2001.

[236] Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. The curious case of hallucinations in neural machine translation, 2021.

[237] Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. The troubling emergence of hallucination in large language models - an extensive definition, quantification, and prescriptive remediations. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 2541–2573, Singapore, December 2023. Association for Computational Linguistics.

[238] Siva Reddy, Danqi Chen, and Christopher D. Manning. CoQA: A conversational question answering challenge. Transactions of the Association for Computational Linguistics, 7:249–266, 2019.

[239] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging NLP models. In Iryna Gurevych and Yusuke Miyao, editors, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 856–865, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[240] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5418–5426, Online, November 2020. Association for Computational Linguistics.

[241] Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. Getting closer to ai complete question answering: A set of prerequisite real tasks. In AAAI Conference on Artificial Intelligence, 2020.

[242] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In Proceedings of the 2018 Conference on Empirical

Methods in Natural Language Processing, pages 4035–4045, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[243] Clemens Rosenbaum, Ignacio Cases, Matthew Riemer, and Tim Klinger. Routing networks and the challenges of modular and compositional computation, 2019.

[244] Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning, 2017.

[245] Rachel Rudinger, Chandler May, and Benjamin Van Durme. Social bias in elicited natural language inferences. In Proceedings of the First ACL Workshop on Ethics in Natural Language Processing, pages 74–79, Valencia, Spain, April 2017. Association for Computational Linguistics.

[246] Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. Masked language model scoring. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2699–2712, Online, July 2020. Association for Computational Linguistics.

[247] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[248] Beatrice Savoldi, Marco Gaido, Luisa Bentivogli, Matteo Negri, and Marco Turchi. Gender bias in machine translation. Transactions of the Association for Computational Linguistics, 9:845–874, 2021.

[249] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4498–4507, Online, July 2020. Association for Computational Linguistics.

[250] Viktor Schlegel, Goran Nenadic, and Riza Batista-Navarro. Semantics altering modifications for evaluating comprehension in machine reading, 2021.

[251] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[252] Sina Semnani, Violet Yao, Heidi Zhang, and Monica Lam. WikiChat: Stopping the hallucination of large language model chatbots by few-shot grounding on Wikipedia. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, Findings of the Association for Computational Linguistics: EMNLP 2023, pages 2387–2413, Singapore, December 2023. Association for Computational Linguistics.

[253] Priyanka Sen and Amir Saffari. What do models learn from question answering datasets? In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2429–2438, Online, November 2020. Association for Computational Linguistics.

[254] Krunal Shah, Nitish Gupta, and Dan Roth. What do we expect from multiple-choice QA systems? In Trevor Cohn, Yulan He, and Yang Liu, editors, Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3547–3553, Online, November 2020. Association for Computational Linguistics.

[255] Sagi Shaier, Kevin Bennett, Lawrence Hunter, and Katharina Kann. Emerging challenges in personalized medicine: Assessing demographic effects on biomedical question answering systems. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi, editors, Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 540–550, Nusa Dua, Bali, November 2023. Association for Computational Linguistics.

[256] Sagi Shaier, Kevin Bennett, Lawrence Hunter, and Katharina von der Wense. Comparing template-based and template-free language model probing. In Yvette Graham and Matthew Purver, editors, Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 766–776, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

[257] Sagi Shaier, Lawrence Hunter, and Katharina Kann. Who are all the stochastic parrots imitating? they should tell us! In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi, editors, Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 2: Short Papers), pages 113–120, Nusa Dua, Bali, November 2023. Association for Computational Linguistics.

[258] Sagi Shaier, Lawrence Hunter, and Katharina von der Wense. It is not about what you say, it is about how you say it: A surprisingly simple approach for improving reading comprehension. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, Findings of the Association for Computational Linguistics ACL 2024, pages 8292–8305, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics.

[259] Sagi Shaier, Lawrence Hunter, and Katharina Wense. Desiderata for the context use of question answering systems. In Yvette Graham and Matthew Purver, editors, Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 777–792, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

[260] Sagi Shaier, Ari Kobren, and Philip Ogren. Adaptive question answering: Enhancing language model proficiency for addressing knowledge conflicts with source citations, 2024.

[261] Sagi Shaier, Ari Kobren, and Philip V. Ogren. Adaptive question answering: Enhancing language model proficiency for addressing knowledge conflicts with source citations. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 17226–17239, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[262] Sagi Shaier, Francisco Pereira, Katharina von der Wense, Lawrence E Hunter, and Matt Jones. More experts than galaxies: Conditionally-overlapping experts with biologically-inspired fixed routing, 2025.

[263] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, Findings of the Association for Computational Linguistics: EMNLP 2023, pages 9248–9274, Singapore, December 2023. Association for Computational Linguistics.

[264] Zhihong Shao and Minlie Huang. Answering open-domain multi-answer questions via a recall-then-verify framework. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1825–1838, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[265] Noam Shazeer, Kayvon Fatahalian, William R. Mark, and Ravi Teja Mullapudi. Hydranets: Specialized dynamic architectures for efficient inference. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8080–8089, 2018.

[266] Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In International Conference on Learning Representations, 2017.

[267] Kim Cheng Sheang and Horacio Saggion. Controllable sentence simplification with a unified text-to-text transfer transformer. In Proceedings of the 14th International Conference on Natural Language Generation, pages 341–352, Aberdeen, Scotland, UK, August 2021. Association for Computational Linguistics.

[268] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context, 2023.

[269] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4222–4235, Online, November 2020. Association for Computational Linguistics.

[270] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation, 2021.

[271] Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage, 2022.

[272] Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. Prompting gpt-3 to be reliable, 2023.

[273] Chenglei Si, Shuohang Wang, Min-Yen Kan, and Jing Jiang. What does bert learn from multiple-choice reading comprehension datasets?, 2019.

[274] Chenglei Si, Ziqing Yang, Yiming Cui, Wentao Ma, Ting Liu, and Shijin Wang. Benchmarking robustness of machine reading comprehension models. In Findings of the Association for

Computational Linguistics: ACL-IJCNLP 2021, pages 634–644, Online, August 2021. Association for Computational Linguistics.

[275] M. V. Simkin and V. P. Roychowdhury. Read before you cite! 2002.

[276] 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.

[277] Aviv Slobodkin, Eran Hirsch, Arie Cattan, Tal Schuster, and Ido Dagan. Attribute first, then generate: Locally-attributable grounded text generation, 2024.

[278] Gizem Sogancioglu, Fabian Mijsters, Amar van Uden, and Jelle Peperzak. Bias in (non)-contextual clinical word embeddings, 2022.

[279] 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, 15(56):1929–1958, 2014.

[280] Weiwei Sun, Hengyi Cai, Hongshen Chen, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. Answering ambiguous questions via iterative prompting. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7669–7683, Toronto, Canada, July 2023. Association for Computational Linguistics.

[281] Mujeen Sung, Jinhyuk Lee, Sean Yi, Minji Jeon, Sungdong Kim, and Jaewoo Kang. Can language models be biomedical knowledge bases? In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4723–4734, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[282] Mujeen Sung, Jinhyuk Lee, Sean Yi, Minji Jeon, Sungdong Kim, and Jaewoo Kang. Can language models be biomedical knowledge bases?, 2021.

[283] Alexey Svyatkovskiy, Shao Kun Deng, Shengyu Fu, and Neel Sundaresan. Intellicode compose: Code generation using transformer, 2020.

[284] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. Multi-granular sequence encoding via dilated compositional units for reading comprehension. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2141–2151, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[285] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science, 2022.

[286] Alberto Testoni and Raffaella Bernardi. "I've seen things you people wouldn't believe": Hallucinating entities in GuessWhat?! In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop, pages 101–111, Online, August 2021. Association for Computational Linguistics.

[287] Clare Thornley, Anthony Watkinson, David Nicholas, Rachel Volentine, Hamid R. Jamali, Eti Herman, Suzie Allard, Kenneth J. Levine, and Carol Tenopir. The role of trust and authority in the citation behaviour of researchers. Inf. Res., 20, 2015.

[288] Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B: Statistical Methodology, 58(1):267–288, 1996.

[289] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision, 2021.

[290] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[291] L. Tunstall, L. von Werra, and T. Wolf. Natural Language Processing with Transformers: Building Language Applications with Hugging Face. O'Reilly Media, 2022.

[292] Konstantinos Tzioumis. Data for: Demographic aspects of first names, 2018.

[293] Neeraj Varshney, Mihir Parmar, Nisarg Patel, Divij Handa, Sayantan Sarkar, Man Luo, and Chitta Baral. Can nlp models correctly reason over contexts that break the common assumptions?, 2023.

[294] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

[295] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

[296] 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, pages 5998–6008, 2017.

[297] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[298] Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, May 2021.

[299] Pancheng Wang, Shasha Li, Haifang Zhou, Jintao Tang, and Ting Wang. Toc-rwg: Explore the combination of topic model and citation information for automatic related work generation. IEEE Access, 8:13043–13055, 2020.

[300] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning deep transformer models for machine translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1810–1822, Florence, Italy, July 2019. Association for Computational Linguistics.

[301] Yicheng Wang and Mohit Bansal. Robust machine comprehension models via adversarial training. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 575–581, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[302] Zhenyi Wang, Li Shen, Le Fang, Qiuling Suo, Tiehang Duan, and Mingchen Gao. Improving task-free continual learning by distributionally robust memory evolution, 2022.

[303] Zikang Wang, Linjing Li, and Daniel Zeng. Knowledge-enhanced natural language inference based on knowledge graphs. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, Proceedings of the 28th International Conference on Computational Linguistics, pages 6498–6508, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[304] 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, 2021.

[305] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[306] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[307] Xiaokai Wei, Shen Wang, Dejiao Zhang, Parminder Bhatia, and Andrew Arnold. Knowledge enhanced pretrained language models: A compreshensive survey, 2021.

[308] Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. ArXiv, abs/1707.06209, 2017.

[309] Liang Wen, Houfeng Wang, Yingwei Luo, and Xiaolin Wang. M3: A multi-view fusion and multi-decoding network for multi-document reading comprehension. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1450–1461, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[310] Janyce Wiebe and Ellen Riloff. Creating subjective and objective sentence classifiers from unannotated texts. In Conference on Intelligent Text Processing and Computational Linguistics, 2005.

[311] Janyce M. Wiebe, Rebecca F. Bruce, and Thomas P. O'Hara. Development and use of a gold-standard data set for subjectivity classifications. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, pages 246–253, College Park, Maryland, USA, June 1999. Association for Computational Linguistics.

[312] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition, 2020.

[313] Jia-Yan Wu, Alexander Te-Wei Shieh, Shih-Ju Hsu, and Yun-Nung Chen. Towards generating citation sentences for multiple references with intent control, 2021.

[314] Fei Xia, Bin Li, Yixuan Weng, Shizhu He, Kang Liu, Bin Sun, Shutao Li, and Jun Zhao. MedConQA: Medical conversational question answering system based on knowledge graphs. In Wanxiang Che and Ekaterina Shutova, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 148–158, Abu Dhabi, UAE, December 2022. Association for Computational Linguistics.

[315] Tong Xiao and Jingbo Zhu. Foundations of large language models, 2025.

[316] Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Unraveling the behavior of large language models in knowledge clashes, 2023.

[317] Xinyu Xing, Xiaosheng Fan, and Xiaojun Wan. Automatic generation of citation texts in scholarly papers: A pilot study. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6181–6190, Online, July 2020. Association for Computational Linguistics.

[318] Haoyun Xu, Runzhe Zhan, Derek F. Wong, and Lidia S. Chao. Let's focus on neuron: Neuron-level supervised fine-tuning for large language model, 2024.

[319] Peng Xu, Davis Liang, Zhiheng Huang, and Bing Xiang. Attention-guided generative models for extractive question answering, 2021.

[320] Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. Megatron-cntrl: Controllable story generation with external knowledge using large-scale language models, 2020.

[321] Jun Yan, Yang Xiao, Sagnik Mukherjee, Bill Yuchen Lin, Robin Jia, and Xiang Ren. On the robustness of reading comprehension models to entity renaming. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 508–520, Seattle, United States, July 2022. Association for Computational Linguistics.

[322] Li Yang, Zhezhi He, Junshan Zhang, and Deliang Fan. Ksm: Fast multiple task adaption via kernel-wise soft mask learning, 2020.

[323] Yi Yang, Wen-tau Yih, and Christopher Meek. WikiQA: A challenge dataset for open-domain question answering. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2013–2018, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[324] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[325] 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, 2018.

[326] Kai Yao, Penglei Gao, Lichun Li, Yuan Zhao, Xiaofeng Wang, Wei Wang, and Jianke Zhu. Layer-wise importance matters: Less memory for better performance in parameter-efficient fine-tuning of large language models, 2024.

[327] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. LinkBERT: Pretraining language models with document links. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8003–8016, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[328] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 535–546, Online, June 2021. Association for Computational Linguistics.

[329] Fei Ye and Adrian G. Bors. Task-free continual learning via online discrepancy distance learning, 2022.

[330] Mohit Bansal Yicheng Wang. Robust machine comprehension models via adversarial training, 2018.

[331] Murat Onur Yildirim, Elif Ceren Gok Yildirim, Ghada Sokar, Decebal Constantin Mocanu, and Joaquin Vanschoren. Continual learning with dynamic sparse training: Exploring algorithms for effective model updates, 2023.

[332] Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. Do large language models know what they don't know?, 2023.

[333] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. 2020.

[334] Sina Zarrieß, Hannes Groener, Torgrim Solstad, and Oliver Bott. This isn't the bias you're looking for: Implicit causality, names and gender in German language models. In Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022), pages 129–134, Potsdam, Germany, 12–15 September 2022. KONVENS 2022 Organizers.

[335] Changchang Zeng, Shaobo Li, Qin Li, Jie Hu, and Jianjun Hu. A survey on machine reading comprehension: Tasks, evaluation metrics and benchmark datasets, 2020.

[336] Changchang Zeng, Shaobo Li, Qin Li, Jie Hu, and Jianjun Hu. A survey on machine reading comprehension: Tasks, evaluation metrics and benchmark datasets, 2020.

[337] Haoyu Zhang, Jingjing Cai, Jianjun Xu, and Ji Wang. Pretraining-based natural language generation for text summarization. In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), pages 789–797, Hong Kong, China, November 2019. Association for Computational Linguistics.

[338] Michael J. Q. Zhang and Eunsol Choi. Clarify when necessary: Resolving ambiguity through interaction with lms, 2023.

[339] Qin Zhang, Shangsi Chen, Dongkuan Xu, Qingqing Cao, Xiaojun Chen, Trevor Cohn, and Meng Fang. A survey for efficient open domain question answering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14447–14465, Toronto, Canada, July 2023. Association for Computational Linguistics.

[340] Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. Tell your model where to attend: Post-hoc attention steering for llms, 2023.

[341] Xuanyu Zhang. MCˆ2: Multi-perspective convolutional cube for conversational machine reading comprehension. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6185–6190, Florence, Italy, July 2019. Association for Computational Linguistics.

[342] Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. Knowledge-grounded dialogue generation with pre-trained language models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 3377–3390, Online, November 2020. Association for Computational Linguistics.

[343] Zheng Zhao, Shay B. Cohen, and Bonnie Webber. Reducing quantity hallucinations in abstractive summarization. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2237–2249, Online, November 2020. Association for Computational Linguistics.

[344] Wanjun Zhong, Yifan Gao, Ning Ding, Yujia Qin, Zhiyuan Liu, Ming Zhou, Jiahai Wang, Jian Yin, and Nan Duan. ProQA: Structural prompt-based pre-training for unified question answering. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4230–4243, Seattle, United States, July 2022. Association for Computational Linguistics.

[345] Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [MASK]: Learning vs. learning to recall. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou,

editors, Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5017–5033, Online, June 2021. Association for Computational Linguistics.

[346] Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions, 2023.

[347] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.

[348] Jianing Zhou and Suma Bhat. Paraphrase generation: A survey of the state of the art. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5075–5086, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[349] Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. Context-faithful prompting for large language models, 2023.

[350] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. Mixture-of-experts with expert choice routing, 2022.

[351] Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. Modifying memories in transformer models, 2020.

[352] Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. Retrieving and reading: A comprehensive survey on open-domain question answering, 2021.

[353] Étienne Fortier-Dubois and Domenic Rosati. Using contradictions improves question answering systems, 2023.