Towards Generalized Open Domain Question Answering Systems

Linqing Liu

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Computer Science

University College London

October 27, 2024

I, Linqing Liu, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Generalization remains a paramount yet unresolved challenge for open-domain question answering (ODQA) systems, impeding their capacity to adeptly handle novel queries and responses beyond the confines of their training data. This thesis conducts a comprehensive exploration of ODQA generalization.

We commence with a meticulous investigation into the underlying challenges. Drawing upon studies on systematic generalization, we introduce and annotate questions according to three categories that measure different levels and kinds of generalization: training set overlap, compositional generalization and novel-entity generalization. When evaluating six popular parametric and non-parametric models, we find non-parametric models demonstrate proficiency with novel entities but encounter difficulties with compositional generalization. Noteworthy correlations emerge, such as a positive association between question pattern frequency and test accuracy, juxtaposed with a strong negative correlation between entity frequency and test accuracy, attributable to closely related distractors. Factors influencing generalization include cascading errors originating from the retrieval component, question pattern frequency, and entity prevalence.

Building on these insights, the focus pivots towards the enhancement of passage retrieval. We propose a novel contextual clue sampling strategy using language models to address the vocabulary mismatch challenge in lexical retrieval for ODQA. This two-step method, comprising filtering and fusion, generates a diverse set of query expansion terms, yielding retrieval accuracy similar to dense methods while notably reducing the index size.

The subsequent phase concentrates on refining reader models in ODQA through flat minima optimization techniques, incorporating Stochastic Weight Averaging (SWA) and Sharpness Aware Minimization (SAM). Rigorous benchmarking underscores the impact of dataset characteristics and model architecture on optimizer effectiveness, with SAM particularly excelling in Natural Language Processing tasks. The combination of SWA and SAM yields additional gains, underscoring the pivotal role of flatter minimizers in fostering enhanced generalization for reader models in ODQA.

Impact Statement

Question answering serves as the vital link connecting human curiosity to the vast potential of machines, guiding exploration through the boundless realm of information. The task of Open Domain Question Answering (ODQA) occupies a prominent position in this endeavor, charged with the responsibility of providing precise answers to textual queries across various domains. The intricacies of ODQA encompass understanding question semantics, retrieving relevant knowledge, and delivering precise answers within the nuanced context of extensive information sources.

This thesis introduces a robust benchmarking framework tailored specifically for evaluating the generalization capabilities of ODQA models. Encompassing different levels of generalization requirements, this framework emerges as an essential resource within the academic community. Researchers and practitioners can use this benchmark to thoroughly evaluate the performance of a wide range of ODQA models, taking into account various dimensions and challenges.

Moreover, the thesis presents innovative strategies to elevate the capabilities of both the reader and retriever components within the ODQA system. In response to practical constraints encountered in real-world deployment, the work focuses on refining the retriever component, resulting in improved accuracy while minimizing index size. This practical enhancement ensures that ODQA models operate more efficiently in real-life scenarios. Concurrently, optimization efforts are directed towards the reader component to strengthen its generalization abilities.

Acknowledgements

I would like to convey my deepest appreciation to my primary supervisor, Pontus Stenetorp, whose unwavering support and generous encouragement have been instrumental in shaping both my research and personal growth. Pontus consistently recognizes even the smallest advancements, creating a positive and motivating environment. During challenging periods when setbacks arose in my research, I vividly remember one instance in a weekly meeting where I expressed frustration over the current progress. Despite my somber mood, Pontus responded with a raised eyebrow, stating, "No way! What you've done is anything but insignificant." His encouragement, paired with remarkable patience, guided me through numerous obstacles, defining the trajectory of this transformative journey.

I am equally grateful for my secondary supervisor, Sebastian Riedel, whose profound wisdom has significantly influenced every facet of our collaboration. Sebastian has provided me with invaluable guidance on research methodologies. Even in casual conversations, his insights have left a lasting impact on my thinking and professional behavior. His remarkable talent for making complex ideas clear consistently guides me in the right direction, making him an indispensable mentor in both my academic and professional journey.

I extend my appreciation to my examiners, Prof. Jonathan Berant and Prof. John Shawe-Taylor, for serving on my graduation committee. Their detailed comments on the thesis and engaging discussions during the viva have greatly enriched my understanding. It is a tremendous honor for me to have their companionship during the final sprint of my PhD.

I express my thanks to the numerous collaborators and friends who enriched my

PhD journey. While the list is extensive, I'd like to name a few in particular: Patrick Lewis, Yuxiang Wu, Pasquale Minervini, Max Bartolo, Jean Kaddour, Maximilian Mozes, Minghan Li, Yihong Chen, Fanghua Ye, Jiayi Wang, David Adelani, Oana-Maria Camburu and Karun Kumar. In the vibrant yet sometimes solitary atmosphere of London, your companionship has been the source of warmth and joy. A heartfelt appreciation goes to my long-term collaborator and dear friend, Ralph Tang. Beyond his evident talent, Ralph's unwavering passion and dedication to research serve as a constant inspiration. Engaging in conversations and collaborative efforts with him is like catching sparks of creativity. This final project of my PhD would not have received the Best Paper award without his invaluable contributions.

I am deeply grateful to my internship mentors, Arthur Mensch and Igor Babuschkin, whose guidance has been truly enlightening. Working alongside them is such an eye-opening experience. Their visionary ideas, profound insights, and exceptional skills in both research and engineering have left a lasting impact on me. As I approach graduation, I am also sincerely thankful to Sam Shah for the immense trust he has placed in me. His encouragement has empowered me to explore my potential, take risks, and tackle challenges with confidence. I extend the same heartfelt appreciation to Miaosen Wang for his invaluable trust and support.

Throughout my stay in London, I cherished exploring cities, working, reading, or simply daydreaming in a cafe. I extend my gratitude to the following cafes for graciously hosting me, providing a space where I could spend entire afternoons in my own corner: Out Of Office Coffee - Wembley, The Hoxton - Holborn, PAUL - Marylebone, and Gecko Coffeehouse - Bethnal Green Road. Here's to the comforting embrace of coffee and the delightful moments spent in these welcoming establishments. May the coffee be with you! :)

While it's already a lengthy list of thank-you, there are a few additional acknowledgments that are crucial. Foremost among them is my heartfelt appreciation to my parents for their enduring love and unwavering support. I also want to give a special shout-out to my best teammate, Tian, who has been my rock and pillar of support throughout. A big thank you extends to the friends I met during my internship, each contributing to the journey in their own unique way. Finally, I want to extend some gratitude to myself, for sticking through and overcoming challenges. I hope to carry forward the courage and confidence I've cultivated for the upcoming challenges. Here's to self-resilience!

Contents

List of Figures

List of Tables

[1]

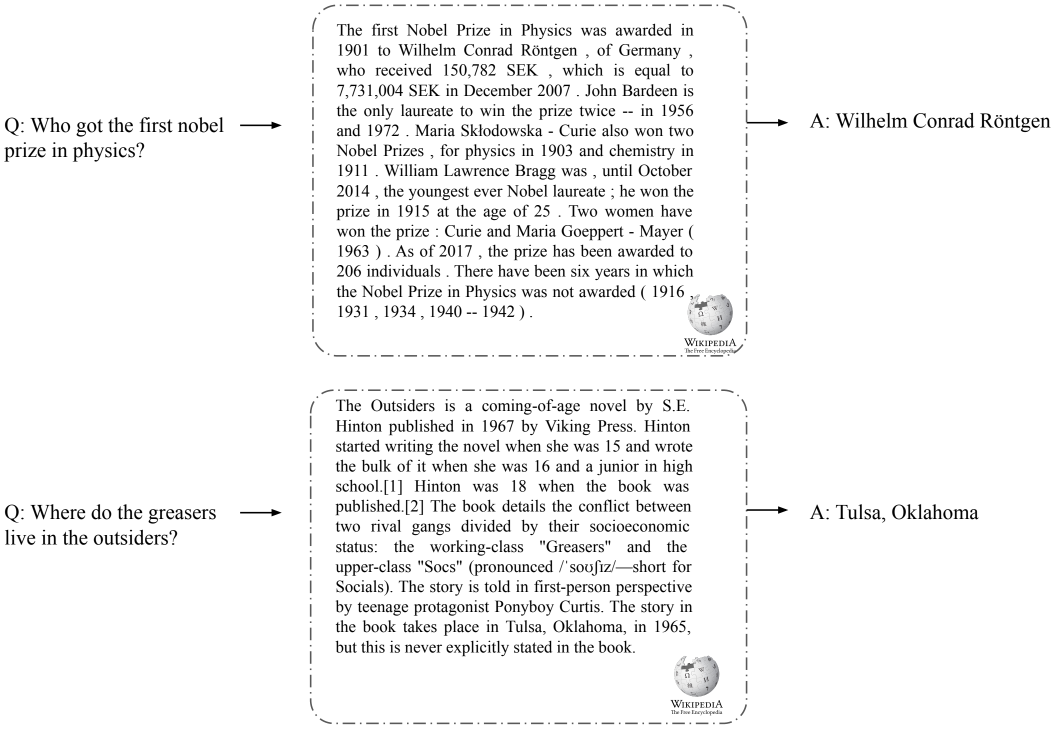
Chapter 1

Introduction

1.1 Thesis Overview

Question answering acts as the intricate bridge connecting human curiosity with the immense potential of machines and the limitless expanse of information. At its core, questions function as a conduit through which humans strive to acquire knowledge about the world surrounding them. Open Domain Question Answering (ODQA) is such a task, where the system aims to provide accurate answers to the textual questions inquiring specific facts across various topics or domains. Considering the example question, as shown in Figure 1.1, the system needs to first understand the semantics of the question, retrieve relevant knowledge to support the facts around the question, then finally decide an answer given all the retrieved contexts.

Comprehending the question itself presents a significant challenge. In the question, “Who got the first Nobel Prize in physics?”, there are several steps involved in both linguistic and semantic analysis. The question word “who” indicates that the answer type is a *NAME* we are seeking. Traditional QA systems heavily rely on the surface textual patterns of the question to classify the question and answer (Mollá et al., 2006; Ravichandran and Hovy, 2002; Voorhees, 2001). The model also needs to identify named entities in the question, such as “Nobel Prize” and “physics”, which are relevant to the context of the question. Additionally, it’s crucial to understand the qualifier “first”, which requires strong reasoning capabilities in models to grasp this specific requirement.



Q: Who got the first nobel prize in physics?

The first Nobel Prize in Physics was awarded in 1901 to Wilhelm Conrad Röntgen , of Germany , who received 150,782 SEK , which is equal to 7,731,004 SEK in December 2007 . John Bardeen is the only laureate to win the prize twice -- in 1956 and 1972 . Maria Skłodowska - Curie also won two Nobel Prizes , for physics in 1903 and chemistry in 1911 . William Lawrence Bragg was , until October 2014 , the youngest ever Nobel laureate ; he won the prize in 1915 at the age of 25 . Two women have won the prize : Curie and Maria Goeppert - Mayer (1963) . As of 2017 , the prize has been awarded to 206 individuals . There have been six years in which the Nobel Prize in Physics was not awarded (1916 , 1931 , 1934 , 1940 -- 1942) .

A: Wilhelm Conrad Röntgen

Q: Where do the greasers live in the outsiders?

The Outsiders is a coming-of-age novel by S.E. Hinton published in 1967 by Viking Press. Hinton started writing the novel when she was 15 and wrote the bulk of it when she was 16 and a junior in high school.[1] Hinton was 18 when the book was published.[2] The book details the conflict between two rival gangs divided by their socioeconomic status: the working-class "Greasers" and the upper-class "Socs" (pronounced /ˈsoʊʃɪz/—short for Socials). The story is told in first-person perspective by teenage protagonist Ponyboy Curtis. The story in the book takes place in Tulsa, Oklahoma, in 1965, but this is never explicitly stated in the book.

A: Tulsa, Oklahoma

Figure 1.1: An example for the task of open domain question answering, where the question is from the Natural Question dataset (Kwiatkowski et al., 2019a), and the document is from Wikipedia titled "List of Nobel laureates in Physics" and "The Outsiders (novel)".

The system then needs to search the vast knowledge sources (e.g. Wikipedia) to retrieve supportive facts regarding to the question. Ideally, the document should exhibit a strong affinity to the question, with the answer seamlessly nestled within an easily recognizable context. For example, in the upper document in Fig. 1.1, the context “The first Nobel Prize in Physics was awarded in 1901 to Wilhelm Conrad Röntgen” is an ideal textual match to the question content “the first Nobel Prize in Physics”. Conversely, “The story in the book takes place in Tulsa, Oklahoma” in the lower figure mandates a more profound understanding to establish the connection between the question’s mention of “greasers” and the context’s reference to the unfolding “story”.

Finally, with the retrieved knowledge from the previous step and the initial processed question, the system will output an answer. This step is also known as Reading Comprehension, where the goal is to identify answers from the paired document. This step can be challenging due to various factors: 1) Complex question structure: sometimes questions are long and complicated, lacking clear question words; 2) Complex document context: the information within the document may require careful reasoning to derive the essence of its content. Additionally, the presence of close but incorrect distractors across different retrieved documents adds complexity; 3) Models may sometimes offer incorrect answers that belong to the same entity type as the correct one. For instance, picking the right name, date, or number among multiple similar mentions in a single document can be difficult.

In this thesis, my focus lies on the enhancement of ODQA generalization. The primary question at hand is: what exactly does "generalization" mean within the context of ODQA? To fully grasp the concept of generalization in ODQA, it's essential to revisit the broader perspective of generalization in the realm of Machine Learning. This concept pertains to a trained model's capacity to perform effectively on novel or unseen data points that were not part of its training phase. Ultimately, the goal of any machine learning model is to understand and generalize patterns from the training data to new, real-world situations. Translating this notion to the specific domain of ODQA, the challenge of generalization encompasses the subsequent dimensions:

1. **Discrepancy in Model Performance between Memorized and Unseen Question-Answer pairs:** There is a large discrepancy in model performance between questions and answers observed at train time and novel questions and answers – even if they are derived from the same distribution (Lewis et al., 2021b).

2. **Over-reliance on Memorization:** During training, models may excessively rely on memorized information, often ignoring pertinent documents even when provided. This tendency to memorize undermines the anticipated ability of the system to generate answers coherent with retrieved information. Consequently, this behavior impairs system interpretability and gives rise to hallucination issues (Bender et al., 2021; Krishna et al., 2021; Longpre et al., 2021).

3. **Temporal and Spatial Generalization:** Current language models are typically

trained on static, overlapping time periods. This likely overestimates their ability for temporal generalization - to perform well on future data (Kasai et al., 2022; Lazaridou et al., 2021; Liska et al., 2022). A similar issue arises with spatial generalization - adapting answers to different geographical contexts (Zhang and Choi, 2021).

4. **Addressing Question Ambiguity:** Ambiguity of the questions. When dealing with ambiguous and unclear questions, it's important to find a set of distinct, equally plausible answers to the question, and provide minimal yet unambiguous rewrites of the question that clarify the interpretation which leads to each answer (Chen et al., 2021a; Keyvan and Huang, 2022; Min et al., 2020)

Subsequently, I will delve into the strategies employed to tackle the challenges associated with generalization in ODQA. The ensuing contents provide a comprehensive breakdown of each chapter's content and its respective description.

PART I: Challenges in Generalization in ODQA

To start with, we need to delve into the factors that contribute to the challenges posed by novel questions. To achieve this, the paper introduces a comprehensive framework that categorizes ODQA generalization into three distinct dimensions: overlap, compositional generalization (comp-gen), and novel-entity generalization. Overlap pertains to test questions that exhibit high lexical similarity with training data, while comp-gen involves questions that creatively combine known facts in novel ways. Lastly, novel-entity questions encompass those containing entities unseen during training.

In pursuit of a comprehensive understanding, we embark on a methodical approach to annotate each question in the current popular datasets. It begins by decomposing questions into elemental components like question words, verbs, and entities. Furthermore, we meticulously annotate questions within three established ODQA datasets — Natural Questions, WebQuestions, and TriviaQA — based on the three identified categories. This groundwork allows for the rigorous evaluation of six diverse ODQA models, ranging from retrieval-based approaches to parametric models, across the specific subsets associated with each category.

The findings presented within this thesis underscore significant performance disparities between the overlap subset and the others. For instance, there emerges a considerable 45.7% performance gap for comp-gen questions within the Natural Questions dataset. Notably, non-parametric models demonstrate a knack for handling novel entities effectively, yet falter when confronted with comp-gen questions. Conversely, parametric models exhibit reduced performance on novel entities. Intriguingly, the accuracy of the models positively correlates with the frequency of question patterns, but this relationship is counterbalanced by a negative correlation between entity frequency and accuracy due to the prevalence of closely related distractors. It also comes to light that, for comp-gen questions, the pertinent context often remains absent within retrieved passages.

The subsequent phase of the study delves into a comprehensive analysis, probing various factors to discern the nuances of model behavior. The investigation encompasses facets like question pattern frequency, passage retrieval, and instances where context is lacking. The collective insights gleaned from this analysis pave the way for the identification of key issues. Suggestions for enhancing passage retrieval, effectively handling missing context, and innovatively combining known facts are among the proposed solutions. Our main contributions are:

1. Pioneering an in-depth investigation into generalization within open-domain question answering (ODQA), utilizing distinct categories to gauge various levels and types of generalization. These categories serve as markers in annotating three previously established ODQA datasets [1].

2. Unveiling a notable revelation: non-parametric models exhibit a relatively proficient handling of novel question entities, yet they grapple with the intricate nuances of compositional generalization.

3. Demonstrate and quantify key factors that impact model generalization perfor-

[1] `https://github.com/likicode/QA-generalize`

mance, which we believe will show the direction for future research towards more robust and generalizable ODQA models.

PART II: *Improving Retriever* - Query Expansion Using Contextual Clue Sampling with Language Models

We delve into the task of sparse retrieval approaches, specifically focusing on the effectiveness of query expansion in addressing the persistent vocabulary mismatch between queries and documents. Despite the emergence of dense retrieval techniques like DPR (Karpukhin et al., 2020) based on semantic matching for open-domain question answering, methods rooted in lexical matching such as BM25 retain significance due to their space efficiency. They also serve as valuable inputs for hybrid approaches (Formal et al., 2021a; Gao et al., 2021). Central to lexical retrieval's challenges is the inherent mismatch in vocabulary between the user's query and the content of documents being retrieved. To mitigate this long-standing issue, query expansion methods, which have a history spanning over half a century (Salton, 1971), have proven effective. Traditionally, these expansion terms are derived from relevant corpora using pseudo-relevance feedback techniques. In recent exploration, the study by GAR (Mao et al., 2021) endeavors to reduce the dependency of query expansion on external corpora. Instead, it employs a large language model to generate context as a substitute. However, a pivotal concern arises in the process of generating expansion terms: achieving a delicate balance between diversity and relevance. Diversity entails capturing multiple possible reasoning paths or contextual clues that lead to the accurate answer for a given question. Relevance, on the other hand, necessitates ensuring that generated contexts align with the semantic context of the query and do not introduce factual errors or semantic irrelevance. A challenge emerges when generating multiple contexts to enhance diversity, as it often results in the generation of incorrect or irrelevant information, a phenomenon known as "hallucination."

To tackle these challenges, we introduce a two-step approach: filtering and fusion. Following the sampling of top-k outputs from the decoder of the fine-tuned language model, the generated contextual clues are grouped into clusters based on their lexical similarity. In each cluster, where similarities are high, a single context with the highest generation probability is retained. This filtration step effectively eliminates potential factual errors and redundant duplicates. Subsequently, the query is individually enriched with each filtered contextual clue, leading to separate document retrievals for every augmented query. In the final step, all retrieved documents are ranked together using the generation probability from the integral contextual clue in the augmented query.

The evaluation of this approach occurs on benchmark datasets Natural Questions (Kwiatkowski et al., 2019a) and TriviaQA (Lee et al., 2019). Comparative analysis reveals that the proposed method bridges the performance gap between baseline models like GAR and dense retrieval models such as DPR. This approach outperforms GAR by 3.1% and 2.9% in terms of Top-5/Top-20 accuracy on the NQ dataset. Furthermore, when compared with DPR, it attains higher Top-100 accuracy by 0.6 and 1.0 points on the two datasets, while remarkably reducing the storage space required for indexing by 96%. Additionally, the improved retrieval performance extends its benefits to downstream question answering tasks, wherein the proposed method enhances the Exact Match scores by 3.2% and 0.8% compared to documents retrieved using the DPR and GAR techniques. Our main contributions:

1. Propose a novel method to generate diverse and relevant contextual clues from a language model to expand queries for lexical retrieval.

2. Filters redundant clues by clustering based on lexical similarity and keeping top ranked per cluster; Augments original query with each filtered clue and fuses retrieved results weighted by generation probability.

3. Our lexical matching based approach achieves a similar top5/top-20 retrieval accuracy and higher top-100 accuracy compared with the well-established dense retrieval model DPR, while reducing the index size by more than 96%.

PART III: *Improving Reader* - Training the reader model with Flat

Minima Optimizers

Previous research has proposed and empirically demonstrated that flat minima within the loss landscape tend to offer superior generalization capabilities compared to sharp minima. These flat minima exhibit enhanced resilience against minor perturbations in input data. Recent studies have introduced techniques such as Stochastic Weight Averaging (SWA) and Sharpness Aware Minimization (SAM), showcasing their ability to locate flatter minima and enhance generalization performance in tasks like image classification and language modeling. Despite this progress, a comprehensive analysis regarding the specific conditions under which these flat minima optimizers prove effective remains limited. Our aim is to delve into the efficacy of two prominent flat minima optimizers—Stochastic Weight Averaging (SWA) and Sharpness Aware Minimization (SAM)—across a diverse array of deep learning tasks. The overarching objective is to offer a meticulous comparison that assists practitioners in selecting the most appropriate optimizer for their specific problem.

Initially, we delve into the geometric characteristics of solutions derived from SWA and SAM by applying them to representative tasks: image classification utilizing WideResNet on CIFAR-100 and code summarization employing Graph Isomorphism Networks on OGB-Code2. Through visual analyses of loss landscapes, our investigation reveals that solutions obtained through SAM occupy distinct basins in comparison to non-flat solutions, situating closer to sharper directions. Interestingly, the fusion of the SWA concept with SAM, particularly through the averaging of SAM iterates, referred to as WASAM, yields the flattest solutions.

Expanding our inquiry, we rigorously benchmark SWA, SAM, and WASAM across an extensive spectrum of 42 tasks encompassing computer vision, natural language processing, and graph representation learning. This comprehensive evaluation accounts for various model architectures, including Convolutional Neural Networks (CNNs), Transformers, and Graph Neural Networks. The tasks span classifications, self-supervised learning, open-domain question answering, natural language comprehension, and graph property prediction. Hyperparameter tuning is consistently executed through validation sets.

Our pivotal findings underscore that the effectiveness of optimizers is influenced by both dataset characteristics and architecture. SWA excels in graph-related tasks, while SAM demonstrates greater prowess in Natural Language Processing (NLP) undertakings. Transformers pose a challenge for SWA, whereas both optimizers yield balanced enhancements in vision-oriented tasks. Intriguingly, the combination of SWA and SAM generates the most consistent and robust improvements across the board. In instances where flat minima optimizers fail to yield improvements, we observe a lack of correlation between training and test loss contours. Focusing on ODQA reader models, in the context of Natural Questions, SAM consistently uplifts performance (+0.33), whereas SWA yields mixed outcomes (-0.20) compared to the baseline. Impressively, averaging SWA and SAM (WASAM) demonstrates the most substantial enhancement (+0.48). Similarly, for tasks like TriviaQA, SAM displays remarkable gains (+0.89), surpassing SWA's improvements (+0.40). Once again, combining SWA and SAM (WASAM) garners the most pronounced benefits (+0.92). The main contributions are:

1. *Comprehensive comparison of minima found by SWA and SAM:* We visualize linear interpolations between different models and quantify the minimizers' flatnesses. This analysis yields 4 insights, e.g., despite SAM finding flatter solutions than SWA as quantified by Hessian eigenvalues, they can be close to sharp directions, a phenomenon that has been overlooked in the previous SAM literature. Averaging SAM iterates leads to the flattest among all minima.

2. *Extensive Performance Evaluation of SWA and SAM:* The performance of SWA and SAM is rigorously evaluated across 42 tasks spanning various domains and model types, such as Computer Vision, Natural Language Processing, and Graph Representation Learning. Nine crucial findings arise, underscoring the impact of dataset and architecture on optimizer effectiveness. Notably, SAM consistently enhances results in NLP tasks, while SWA is more effective for GRL tasks. The study's code and hyperparameters [2] are openly accessible, fostering reproducibility and further research.

[2] https://github.com/JeanKaddour/WASAM

1.2 Previously Published Materials

This thesis incorporates experiments and discoveries stemming from papers that have been previously published. These papers are enumerated below. Any contributions not originating from the author of this thesis are clearly indicated at the commencement of the respective chapters.

- **Linqing Liu**, Patrick Lewis, Sebastian Riedel, and Pontus Stenetorp. "Challenges in Generalization in Open Domain Question Answering." In *Findings of the Association for Computational Linguistics: NAACL 2022, pp. 2014-2029. 2022.*

- **Linqing Liu**, Minghan Li, Jimmy Lin, Sebastian Riedel, and Pontus Stenetorp. "Query Expansion Using Contextual Clue Sampling with Language Models." *arXiv preprint arXiv:2210.07093 (2022).*

- *Jean Kaddour, ***Linqing Liu**, Ricardo Silva, and Matt J. Kusner. "When do flat minima optimizers work?." Advances in *Neural Information Processing Systems 35 (2022): 16577-16595.* (*Equal Contribution)

- *Raphael Tang, ***Linqing Liu**, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Jimmy Lin, and Ferhan Ture. "What the daam: Interpreting stable diffusion using cross attention." *ACL 2023* (*Equal Contribution).

- Patrick Lewis, Yuxiang Wu, **Linqing Liu**, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. "Paq: 65 million probably-asked questions and what you can do with them." Transactions of the Association for Computational Linguistics 9 (2021): 1098-1115.

Chapter 2

Background

In this Chapter, I'll provide an overview of the Open Domain Question Answering (ODQA) task. In Section 2.1, I'll delve into the fundamental task definition and explore its historical context, tracing the evolution of traditional methodologies. Subsequently, I will explore the latest research trends, focusing on three primary approach paradigms: Retriever-Reader (Section 2.2), QA-pair retriever (Section 2.3), and Parametric models (Section 2.4). The first two paradigms are often referred to as non-parametric methods in certain contexts due to their reliance on external knowledge for answer prediction, while parametric models operate solely based on their inherent model parameters. In Sec.2.5, I will introduce the most widely-used approaches to evaluate the retriever and reader model.

2.1 Task Formulation and History

The task of Open Domain Question Answering (ODQA) aims to provide answers to the input questions of a diverse range of topics, without given any specific relevant context. It requires the system to first explore relevant knowledge (either externally or internally) in order to answer the question. The underlying sources of information that enable question answering encompass various forms, including structured knowledge bases, semi-structured tabular data, and unstructured textual content extracted from web documents. In this thesis, we will focus on the factoid natural language questions, expecting the answers to be short and concise, and the knowledge source to be textual documents from a large corpus such as Wikipedia.

The exploration of Open Domain Question Answering (ODQA) dates back to a rich history of research endeavors. One of the earliest and widely recognized question-answering systems was crafted to address inquiries related to American baseball games, as documented by Green Jr et al. (1961). This pioneering system was composed of two core components: linguistic and semantic analyses. The initial phase, termed the "syntactic routine," undertook a sequential identification of noun phrases, prepositional phrases, and adverbial phrases, consequently pinpointing the subject and object linked to each verb. Additionally, it scrutinized words designated as question words. To illustrate, consider the question, "How many games did the Yankees play in July?" The syntactic analysis transformed this query into a structured form, "[How many games] did [the Yankees] play (in [July])?" Each annotation carried distinct meanings, aiding subsequent stages. Subsequently, the "Content Analysis" phase harnessed dictionary meanings and the outcomes of the syntactic analysis to formulate a specification list tailored for the processing program. In the example mentioned earlier, the phrase "how many games" transformed into the directive "$\text{Game}_{(\text{number of})} = ?$". Lastly, the processor endeavored to locate matches for all the specified pairs on the generated spec list. This early work exemplified the foundational steps towards automating question-answering through linguistic and semantic analysis, marking a significant milestone in the journey towards more sophisticated ODQA systems. However, they heavily depend on manually crafted rules for linguistic and semantic analysis, struggle with variations in question phrasing and structure, and specifically tailored to some subsets of English questions (Bobrow et al., 1964; Kirsch, 1964; Simmons, 1965).

The concept of ODQA emerged within the context of the QA track is initiated by the Text Retrieval Conference (TREC-8; Voorhees et al., 1999). The goal of the track is to foster research on systems that retrieve answers rather than documents in response to a question, with an emphasis on systems that can function in unrestricted domains. Each response was evaluated by human assessors, who considered a response correct if it contained an answer within the snippet. However, the comparative effectiveness of different systems was often obscured by the situation where two snippets could both contain correct answers, yet one was markedly superior as a response (Ellen, 2001). To address this issue and encourage systems to exhibit their capability to precisely locate answers, the TREC 2002 task (Voorhees, 2003) demanded systems to provide exact answers—complete answer text without any additional content. Responses that included a correct answer along with extraneous text were considered "inexact" and did not contribute to a system's score. Pinpointing the exact boundary of an answer was a more challenging problem compared to merely identifying a text portion containing an answer. Additionally, certain applications of QA technology did not necessitate this additional step. In light of this, the TREC 2003 track (Voorhees and Buckland, 2003) introduced a "passages" task to cater to research groups interested in applications that allowed for the return of text segments containing answers. Simultaneously, the main task continued to require exact answers. While the passages task's question set exclusively contained factoid questions, the main task encompassed a broader range, including list and definition questions alongside factoid questions. These different question types were evaluated separately, and the final score for a main task run was a composite of the scores attained for the three distinct question types. Since TREC 2004 (Voorhees, 2004), factoid and list questions were organized into distinct series, each focused on a specific target. This format enabled the evaluation of different question types while simulating a user session. Targets could be people, organizations, or things. In TREC 2005 and 2006 (Voorhees and Dang, 2005), answers are required to be temporally accurate within the series' timeframe, and the distinction between locally and globally correct answers was reinforced. They also introduce a refined nugget pyramid evaluation method, in which multiple assessors provide judgments of whether a nugget was vital or simply okay. In TREC 2007 (Dang et al., 2007), the series-based question format continued, but the document collection expanded to include both newswire and blogs. This introduced challenges in handling informal and less reliable discourse structures and non-well-formed language, crucial for real-world QA systems. Overall, the progression of the TREC Question Answering track aims to closely align with real-world challenges, significantly propelling the

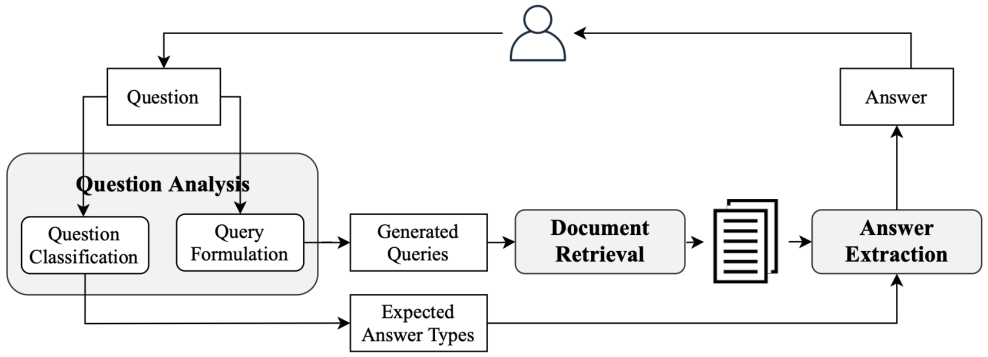


Figure 2.1: An illustration of traditional architecture of the ODQA system (Zhu et al., 2021).

advancement of ODQA.

The traditional ODQA system comprises three steps: Question Analysis, Document Retrieval, and Answer Extraction (illustrated in Fig.2.1). I will discuss each of the step in the follows:

Question Analysis. It plays a crucial role in understanding and preparing the user's question for subsequent stages. It usually involves a *query formulation* module, where the question is processed into syntactic parse structures, and semantic key words (Fan et al., 2005; Kwok et al., 2001). The question's syntactic structure could better help extract plausible answers from the pages returned by the search engine. Moreover, this process leans on the *question classification* module to ascertain the specific type of question based on a predefined classification system. Questions generally adhere to predictable linguistic patterns, enabling their classification into categories like "what," "why," "who," "how," "when," "where," and more. The determined question type then significantly reduces the pool of possible answers during the answer extraction stage (Allam and Haggag, 2012). For instance, consider the question "Q: Which city has the largest population?" The objective here is to classify this question within the "city" answer type, which implies that only responses fulfilling the criteria of being cities need to be considered. Notably, Li and Roth (2002) define a two-layered taxonomy, which represents a natural semantic classification for typical answers in the TREC task. The hierarchy contains 6 coarse classes (ABBREVIATION, ENTITY, DESCRIPTION, HUMAN, LOCATION and NUMERIC VALUE) and 50 fine classes (such as "animal", "body", "color" etc. for ENTITY). They design design a sequence of two simple classifiers to assign questions into fine classes.

Document Retrieval. This step aims at retrieve relevant documents from large corpus that are more likely to contain the answer to the question. Traditional retrieval approaches include Boolean Model, Vector Space model and Probabilistic Models (Schütze et al., 2008). The *Boolean Model* is a simple and foundational retrieval approach. In this model, both the queries and documents are represented as sets of terms (words). The idea is to express queries using Boolean operators like AND, OR, and NOT to combine terms and retrieve documents that match the query's terms. If a document contains all the terms in the query, it's considered relevant. Conversely, documents not containing certain terms are excluded using NOT. The *Vector Space Model* represents both queries and documents as vectors in a high-dimensional space. Each dimension corresponds to a unique term, and the vector's value in each dimension represents the term's importance in the document or query (Salton, 1972). The similarity between a query vector and a document vector is computed, often using techniques like cosine similarity. The higher the similarity, the more relevant the document is to the query. This model is more flexible and can capture partial relevancy, allowing for a more nuanced approach to retrieval compared to the Boolean Model. A popular example of *Probabilistic Models* is the Okapi BM25 model (Crestani et al., 1998; Roberts et al., 2020; Robertson et al.). These models integrate probabilistic relationships between terms and documents into their framework. They consider factors like term frequency, document length, and collection statistics to estimate relevance more effectively. BM25 has been particularly successful in this regard, making it a widely used retrieval model.

Answer Extraction. This step fulfils the ultimate goal of the task of ODQA - returning an accurate answer to the given question. Following the question analysis step, the anticipated answer's type is established and aligned with a set of entity categories. Subsequently, Named Entity Recognition (NER) is employed to isolate the entity categories found within a text segment. If a given text fragment lacks entities that match the expected answer's type, the text is either discarded or subjected to significant penalties. Mollá et al. (2006) show that employing multiple labels in NER to enhance the recall of named entities is beneficial for the QA process. When NER provides multiple potential labels for a string or its part, presenting the most credible alternatives aids the QA system in finding answers more effectively. While this approach may introduce noise due to potentially incorrect entities being included, the increase in recall effectively balances out this drawback, making it a valuable strategy for improving QA performance. Moreover, another effective approach explores the potential of surface patterns (Soubbotin and Soubbotin, 2001; Voorhees, 2001). The main idea is: "The core of our question-answering mechanism is searching for predefined patterns of textual expressions that may be interpreted as answers to certain types of questions. The presence of such patterns in analyzed answer-string candidates may provide evidence of the right answer." (Soubbotin and Soubbotin, 2001). Specific answer types are often associated with distinct phrases. For instance, BIRTHDATE questions often yield responses like "Mozart was born in 1756" or "Gandhi (1869–1948)...". This implies that regular expressions such as "<NAME> was born in <BIRTHDATE>" can effectively identify correct answers. Various work present an approach for automatically learning such regular expressions (along with determining their precision) from the web, for given types of questions (Ravichandran and Hovy, 2002).

2.2 Retriever-Reader

The retriever-reader framework includes two modules: a retriever to retrieve the most relevant documents from an extensive corpus in response to a given input query,

	Dense	Sparse
Supervised	DPR (Karpukhin et al., 2020), ANCE (Xiong et al., 2020), PAIR (Ren et al., 2021), AR2 (Zhang et al., 2021), RocketQA (Du et al., 2021), COLBERT (Khattab and Zaharia, 2020)	DeepCT (Dai and Callan, 2020), DeepImpact (Mallia et al., 2021), COIL (Gao et al., 2021), uniCOIL (Lin and Ma, 2021a), SPLADE (Formal et al., 2021a,b)
Unsupervised	LSI (Atreya and Elkan, 2011), LDA (Wei and Croft, 2006)	BM25 (Robertson et al.), tf-idf (Salton et al., 1975)

Table 2.1: A taxonomy of retrieval models (Lin, 2022)

.

and a reader to yield the answer given both the query and the retrieved documents. In this section, we will delve into each module separately: first, exploring the retrievers under a unified taxonomy, followed by an in-depth examination of various approaches to the reader model.

2.2.1 Document Retriever

Information retrieval is commonly defined as follows: When presented with an information need in the form as a query q, the main goal of information retrieval is to provide a ranked list of k documents $d_1, d_2, ..., d_k$ from an extensive yet finite collection of documents D. The task is sometimes also referred to as top-k retrieval (or ranking), with "k" representing the pre-defined length of the ranked list. The top-k ranked lists are usually phrased as the ranked list of results (or the "hits").

Lin, 2022; Lin and Ma, 2021b propose a conceptual framework (as shown in Table 2.1) that unites the traditional sparse retrieval and more recent dense retrieval approaches. They are categorized along two dimensions: the contrast between dense vs. sparse vector representations, and the contrast between supervised (learned) vs. unsupervised approaches. In this section, we will describe the each of retrieval direction under this framework.

Learnt Dense Representations. Dense retrieval method involves training encoders (commonly transformer-based) to convert both queries and documents into dense fixed-width vectors. These dense vectors are designed to capture the semantic meaning and contextual information of the queries and documents. Then the system computes the similarity between the query vector and each document vector. Various similarity metrics like cosine similarity or dot product can be used to measure how closely related the query and documents are. With the calculated similarity scores, the documents are then ranked based on their relevance to the query. The top-k documents with the highest similarity scores are then returned as the search results.

DPR (Karpukhin et al., 2020) is one of the most popular dense retrieval model. They train two independent BERT (Devlin et al., 2019) networks to encode query and document separately, then take the representation at the front CLS token as the output representation. In this way, each document in the collection and each query can be indexed into the low-dimensional and continuous space. During training time, the embedding is designed to optimize the inner product between question and relevant document vectors, following an objective that involves comparing all pairs of questions and documents within a batch. They optimize the loss function by using the negative log likelihood of the positive document:

$$L(q_i, d_i^+, d_{i,1}^-, ..., d_{i,n}^-) = -\log \frac{e^{\mathrm{sim}(q_i, d_i^+)}}{\mathrm{sim}(q_i, d_i^+) + \sum_{j=1}^{n} e^{\mathrm{sim}(q_i, d_{i,j}^-)}} \tag{2.1}$$

where d_i^+ indicates relevant positive document, and d_i^- indicates irrevelant negative document. The selection of in-batch negative examples is decisive for learning a high-quality encoder. Many other research work explores more effective ways of "hard negative" selection. The approach of ANCE (Xiong et al., 2020) constructs negatives from an Approximate Nearest Neighbor (ANN) index of the corpus, which is parallelly updated with the learning process to select more realistic negative training instances. They use a recent checkpoint to update the representation of documents in the corpus and once finished, refreshes the ANN index with most up-to-date encodings. To address the challenge of the existence of unlabeled positives and limited training data, Du et al., 2021 proprose the denoised hard negatives technique which allow to filter false negatives from sampled hard negatives, and data augmentation to increase the amount of training data by adding newly created synthetic data from existing data. Besides the distance between query and document embeddings, the distance between positive and negative document embeddings are also considered (Ren et al., 2021). They learn passage-centric similarity relation for enhancing the dual-encoder architecture through a two-stage training procedure: pre-train the model with combined loss, and fine-tune the model to optimize the query-centric loss only. Other techniques also leverage adversarial training (Zhang et al., 2021) to optimize a ranker and a retriever a minimax adversarial objective. The retriever's objective is to retrieve negative documents strategically, aiming to deceive the ranker. Conversely, the ranker's goal is to rank a set of candidates comprising both ground-truth and retrieved documents. It also provides constructive feedback to the dual-encoder retriever in a progressive manner. Pre-training large bi-encoder models on a corpus of 65 million synthetically generated question-answer pairs from Wikipedia and a corpus of 220 million post-comment pairs from Reddit has also been proven effective (Oguz et al., 2022).

The approaches mentioned earlier all revolve around bi-encoder architectures. However, there exists an alternative design known as cross-encoders, exemplified by the monoBERT model (Nogueira et al., 2019a). In this paradigm, a query and a document are inputted jointly into a pretrained transformer using a specific template. The contextual representation of the [CLS] token is then harnessed for relevance classification. Comparing bi-encoders and cross-encoders, the former enjoys increased efficiency gains because the representations for candidate documents can be precomputed. By employing an approximate nearest neighbor search (ANN), relevant documents can be computed for a query with heightened efficiency. However, cross-encoder approaches consistently outperform bi-encoders due to their adeptness at capitalizing on the relevance attention signals between the query and candidate documents in each layer of the Transformer encoder. A significant breakthrough in this domain is demonstrated by COLBERT (Khattab and Zaharia, 2020), which showcases that ranking methods reliant on dense representations can reach levels of effectiveness comparable to cross-encoder designs. COLBERT takes advantage of the *same* BERT model for encoding both queries and documents. The relevance determination takes place through a subsequent interaction step (*MaxSim* operation), which captures the nuanced similarity between query and document embeddings. This permits the offline precomputation of document representations while maintaining robust matching capabilities. Empirical evaluations underscore that in terms of query latency, COLBERT substantially narrows the gap between monoBERT and pre-BERT neural ranking models, with only modest declines in effectiveness. However, alongside its strengths, COLBERT introduces a concept termed "space efficiency" — the storage space required for document representations. Remarkably, they necessitate 156 GB of storage space for corpus storage, a figure two orders of magnitude larger than the 2.5 GB demanded by the index of the same collection in Lucene. Consequently, when evaluating efficiency, factors such as query latency, computational expenses during the indexing phase, and also storage costs become pivotal considerations.

Unsupervised Dense Representations. There is little work in this category in general. Suggested by Lin, 2022, LSI (Atreya and Elkan, 2011) and LDA (Wei and Croft, 2006) can be included in this category.

Unsupervised Sparse Representation. In the sparse retrieval approach, the query and document is represented as separate sparse bag-of-words vector representations of dimension V (V is the vocabulary size). Each dimension in the sparse vector corresponds to a term in the vocabulary, and each termis weighted according to the BM25 scoring function. More details are described in Sec.4.1.

Learnt Sparse Representations. Sparse representation learning for queries and documents has a long history. This problem can be formulated as a supervised learning task to determine term weights for vectors of the same size as the vocabulary. One of the earliest instances of this concept use genetic algorithms on boolean vectors and a small set of relevance judgments for representational learning (Gordon, 1988). Trotman, 2005 develops better BM25-like scoring functions based on genetic programming.

With the development of pre-trained language models, DeepCT (Dai and Callan, 2020) first generate the contextualized word representations using BERT (Devlin et al., 2019), then predict term weights through linear regression. Given the ground truth term weight for every word in text, DeepCT aims to minimize the mean square error (MSE) between the predicted weights and the target weights. DeepCT has a limitation where it is trained using a per-token regression task, requiring ground truth term weights for each word, which hinders the individual impact scores from co-adapting to the objective of identifying relevant documents. In constrast, DeepImpact (Mallia et al., 2021) addresses this limitation by directly optimizing the sum of query term impacts to maximize the score difference between relevant and non-relevant passages for the query. They aim to jointly learn the final term impact across all query terms occurring in a passage. Additionally, they incorporate the idea from DocT5Query (Nogueira and Cho, 2019) to enrich the document with potential query terms, aiming to overcome the vocabulary mismatch problem. Concurrently, SOIL (Gao et al., 2021) first produce representations for each document token with deep LLM offline, then build the token's contextualized inverted list. During inference time, they use each query token to look up its own inverted list and compute vector similarity with document vectors stored in the inverted list as matching scores. In SOIL, the scoring model assigns each term a vector weight, stored in standard inverted lists. Lin and Ma, 2021a further reduce the token dimension of COIL to one, and degenerate the model into producing scalar weights. Similarly, SPLADE (Formal et al., 2021b) and SPLADE (Formal et al., 2021a) produce a sparse vocabulary-level vector that retains the term-level decomposition of late interaction while simplifying the storage into one dimension per token.

Retrieve-and-Rerank . After the initial retrieval step, the system has already retrieved a set of candidate documents that are potentially relevant to the given query. However, the retrieved documents might still contain irrelevant or noisy information. It is common to employ a later-stage document reranker to reevaluate and refine the ranking of these candidate passages to identify the most relevant ones more accurately. This stage is crucial because the initial retrieval might have retrieved a large number of documents. Its primary objective is to prioritize the most relevant documents, potentially discarding irrelevant ones, and passing only the most promising ones to the subsequent stages. To enhance the effectiveness of this process, the system can deploy multiple rerankers, allowing the output of each stage to serve as the input for the next, leading to a cascade effect. This approach is often referred to as "retrieve-and-rerank." To exploit the tradeoff between effectiveness and efficiency, early reranking stages leverage computational "cheap" features to discard easily identifiable irrelevant documents. Subsequently, the later stage reranking steps focus on computing "expensive" features, but this is done on a relatively smaller subset of candidate documents.

In post-BERT era, Nogueira and Cho, 2019 first propose a two-stage pipeline, where first using BM25 lexical retrieval to fetch documents candidates, then reranking the candidates with BERT neural models. The sentence-level (Yilmaz et al., 2019) and passage-level (Dai and Callan, 2019) relevance scores from BERT are both separately explored for document reranking. Moving beyond simple passage score aggregation strategies, Li et al., 2020b extensively compare the previously proposed approaches for aggregating passage-level signals, and explore strategies for aggregating relevance signals from a document's passages into a final ranking score (PARADE). Other works (Soldaini and Moschitti, 2020) observe that models such as monoBERT (Nogueira and Cho, 2019) works similarly as a multi-stage ranking architecture if we consider each layer of the transformer encoder as a separate ranking stage. With Cascade Transformer, they adapt transformer-based models into a cascade of rankers. Each ranker is used to prune a subset of candidates in a batch, thus dramatically increasing throughput at inference time. Yet more recent work explores merging the two-stage pipeline into an end-to-end dense retrieval task, training a single BERT-based model that is capable of retrieving and rank documents simultaneously (Karpukhin et al., 2020).

2.2.2 Reader

The reader model aims at inferring the answer from a set of retrieved (and reranked) documents in response to the given question. The reader models can be broadly categorized into extractive and generative ones.

Extractive Reader. In brevity, extractive reader extracts a span from the document as an answer to the input question, with the goal of predicting the span start and end positions. The encoder of the reader model initially takes the concatenation of the question and document as the input context, then produces the contextual representations $h_1, h_2, .., h_n$ where h_i corresponds to each of the token in the context. The probability of a token being the start or ending positions of an answer span is calculated as following:

$$\begin{aligned} P_{start,i}(s) &= \text{softmax}\left(h_i \cdot w_{start}\right) \\ P_{end,i}(s) &= \text{softmax}\left(h_i \cdot w_{end}\right) \end{aligned} \tag{2.2}$$

where w_{start} and w_{end} indicates the separate linear layer on top of the contextual representations to independently predict the probability of each context token being start and end positions. The BERT-based extractive readers are widely used in previous work to locate the answer (Karpukhin et al., 2020; Wu et al., 2020; Yang et al., 2019).

Generative Reader. The generative reader is typically based on a sequence-to-sequence model (such as T5; Raffel et al., 2020 and BART; Lewis et al., 2020a). It takes the question and the document as input, and generate the answer. The training objective is to optimize the cross-entropy loss between the predicted answer sequence and the ground-truth answer. For instance, as an initial step, Fusion-in-Decoder (FiD; Izacard and Grave, 2021) encodes the concatenation of the question, each retrieved document and its title into separate hidden state. Then the decoder performs attention over the concatenation of all the resulting representations of all the retrieved documents. The model only fuses all the supported evidence in the decoder. Assuming that the attention scores over question-answer pairs in the reader module effectively indicate the relevance of a passage for answering a question, they delve deeper into the concept. They propose to use the reader module as the teacher model and harnessing the knowledge distilled from the question-answer pairs. Subsequently, they train the retriever to estimate the reader's attention scores (Izacard and Grave, 2020). Retrieval-Augmented Generation (RAG; Lewis et al., 2020b) combine the retriever and the reader in an end-to-end probabilistic model. For the encoded query, they retrieve the top-K most relevant documents and retreat the document representations as a latent variable. The reader model then condition on these latent documents to generate the output. Singh et al., 2021 presents another end-to-end differentiable training method for retrieval-augmented systems that combine information from multiple retrieved documents when generating answers. They approximate the process of marginalizing over retrieved documents with an expectation-maximization algorithm.

While both extractive and generative reader models are successfully applied to the question answering task, it naturally raises a question: what are the strengths and weaknesses of these two category of models? Luo et al., 2022 perform a system comparison on this issue by exploring nine transformer-based models as backbone architectures. Several insightful findings emerged from the comparison: i) The choice of pre-trained language model (T5; Raffel et al., 2020 vs. BART; Lewis et al., 2020a) affects extractive and generative performance; ii) With the extractive reader built on T5's encoder, extractive readers perform better than generative readers on average; iii) Extractive readers excel in short contexts and demonstrate better generalization on out-of-domain datasets and rare answers, whereas generative readers perform better in long contexts; iv) Encoder-decoder models' encoders are effective extractive readers. Extractive readers built on top of BART or T5 encoders outperform encoder-only models such as RoBERTa; v) Longer inference length has a positive effect for all models.

2.3 QA-pair retriever

The material in this section first appeared in:

> Patrick Lewis, Yuxiang Wu, **Linqing Liu**, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. "Paq: 65 million probably-asked questions and what you can do with them." Transactions of the Association for Computational Linguistics 9 (2021): 1098-1115.

Besides the retrieve-and-read paradigm where they typically index the whole corpus, there is another paradigm where models explicitly retrieve (training) QA pairs (Lewis et al., 2021b; Xiao et al., 2021). These models have a number of useful properties, such as fast inference, interpretable outputs (by inspecting retrieved

QA-pairs), and the ability to update the model's knowledge at test time by adding or removing QA-pairs. However, the traditional QA-pair retriever models are currently not competitive with retrieve-andread systems in terms of accuracy, largely because the training QA-pairs they operate on cover substantially less knowledge than background corpora like Wikipedia.

In Lewis et al., 2021c, we present Probably Asked Questions (PAQ), a semi-structured Knowledge Base (KB) of 65M natural language QA-pairs, which models can memorise and/or learn to retrieve from. PAQ differs from traditional KBs in that questions and answers are stored in natural language, and that questions are generated such that they are likely to appear in ODQA datasets. PAQ is automatically constructed using a question generation model and Wikipedia. I will discuss in details about the generation process below.

2.3.1 Generating Question-Answer Pairs

Given a large background corpus C, the QA-pair generation process consists of the following components:

1. A *passage selection* model $p_s(c)$, to identify passages which humans are likely to ask questions about.
2. An *answer extraction* model $p_a(a|c)$, for identifying spans in a passage that are more likely to be answers to a question.
3. A *question generator* $p_q(q|a,c)$ that, given a passage and an answer, generates a question.
4. A *filtering* QA model $p_f(a|q,C)$ that generates an answer for a given question. If an answer generated by p_f does not match the answer a question was generated from, the question is discarded. This ensures generated questions are *consistent*.

As shown in Figure 2.2, these models are applied sequentially to generate QA-pairs, similarly to *contextual* QA generation (Alberti et al., 2019; Lewis et al., 2019). First a passage c is selected with a high probability under p_s. Next, candidate answers a are extracted from c using p_a, and questions q are generated for each answer using

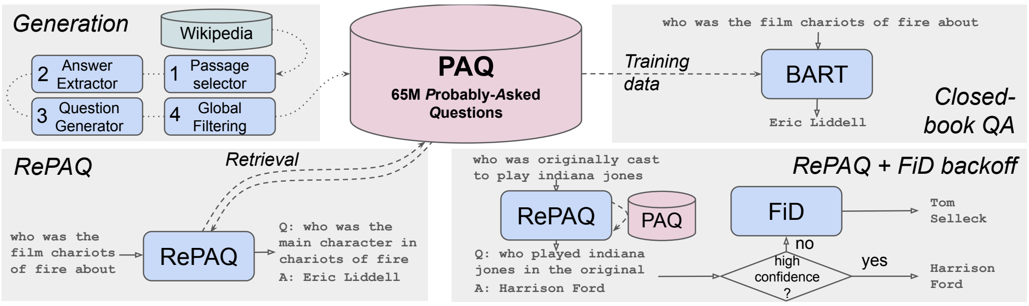


Figure 2.2: Top Left: Generation pipeline for QA-pairs in PAQ. Top Right: PAQ used as training data for CBQA models. Bottom Left: RePAQ retrieves similar QA-pairs to input questions from PAQ. Bottom right: RePAQ's confidence is predictive of accuracy. If confidence is low, we can defer to slower, more accurate systems, like FiD.

p_q. Lastly, p_f generates a new answer a' for the question. If source answer a matches a', then (q,a) is deemed consistent and added to PAQ. The pipeline is based on Alberti et al. (2019), updated to take advantage of recent modelling advances. Passage selection and our filtering approach are novel contributions to the best of our knowledge, specifically designed for ODQA QA-pair generation.

Passage Selection. It is used to find passages that are likely to contain information that humans may ask about, and thus make good candidates to generate questions from. We learn p_s using a similar method to Karpukhin et al. (2020). Concretely, we assume access to a set of positive passages $C^+ \subset C$, obtained from answer-containing passages from ODQA train sets. As we do not have a set of labelled negatives, we sample negatives either randomly or using heuristics. We then maximize log-likelihood of positive passages relative to negatives. We implement p_s with RoBERTa (Liu et al., 2019) and obtain positive passages from Natural Questions (NQ, Kwiatkowski et al., 2019a). We sample *easy negatives* at random from Wikipedia, and *hard negatives* from the same Wikipedia article as the positive passage. Easy negatives help the model to learn topics of interest, and hard negatives help to differentiate between interesting and non-interesting passages from the same article.

Answer Extraction. Given a passage, this component identifies spans that are likely to be answers to questions. We consider two alternatives: an off-the-shelf Named

Entity Recogniser (NER) or training a BERT (Devlin et al., 2019) answer extraction model on NQ.

Question Generation. Given a passage and an answer, this model generates likely questions with that answer. To indicate the answer and its occurrence in the passage, we prepend the answer to the passage and label the answer span with surrounding special tokens. We train on a combination of NQ, TriviaQA, and SQuAD, and perform standard fine-tuning of BART-base (Lewis et al., 2020a) to obtain p_q.

Filtering. The filtering model p_f improves the quality of generated questions, by ensuring that they are *consistent*: that the answer they were generated is likely to be a valid answer to the question. Previous work (Alberti et al., 2019; Fang et al., 2020) has employed a machine reading comprehension (MRC) QA model for this purpose, $p_f(a|q,c)$, which produces an answer when supplied with a question *and* the passage it was generated from. We refer to this as *local filtering*. However, local filtering will not remove questions which are ambiguous (Min et al., 2020), and can only be answered correctly with access to the source passage. Thus, we use an ODQA model for filtering, $p_f(a|q,C)$, supplied with only the generated question, and *not* the source passage. We refer to this as *global filtering*, and later show it is vital for strong downstream results. We use FiD-base with 50 passages, trained on NQ (Izacard and Grave, 2021).

2.3.2 RePAQ Retriever and Reranker

RePAQ Retriever. Our retriever adopts the dense Maximum Inner Product Search (MIPS) paradigm, that has recently been shown to obtain state-of-the-art results in a number of settings (Karpukhin et al., 2020; Lee et al., 2021, inter alia). Our goal is to embed queries q and indexed items d into a representation space via embedding functions g_q and g_d, so that the inner product $g_q(q)^\top g_d(d)$ is maximised for items relevant to q. In our case, queries are questions and indexed items are QA-pairs (q',a'). We make our retriever symmetric by embedding q' rather than (q',a'). As such, *only one* embedding function g_q is required, which maps questions to embeddings. This applies a useful inductive bias, and we find that it aids stability during training.

Learning the embedding function gq is complicated by the lack of labeled question pair paraphrases in ODQA datasets. We propose a latent variable approach similar to retrieval-augmented generation (Lewis et al., 2020b). Once the embedder g_q is trained, we build a test-time QA system by embedding and indexing a QA KB such as PAQ. Answering is achieved by retrieving the most similar stored question, and returning its answer. The matched QA-pair can be displayed to the user, providing a mechanism for more interpretable answers than CBQA models and many retrieve-and-read generators which consume thousands of tokens to generate an answer. Efficient MIPS libraries such as FAISS (Johnson et al., 2019) enable RePAQ's retriever to answer 100s to 1,000s of questions per second. We use a KB for RePAQ consisting of train set QA-pairs and QA-pairs from PAQ.

RePAQ Reranker. Accuracy can be improved using a reranker on the top-K QA-pairs from the retriever. The reranker uses cross-encoding, and includes the retrieved answer in the scoring function for richer featurisation. The model is trained as a multi-class classifier, attempting to classify a QA-pair which answers a question correctly against K-1 retrieved QA-pairs which do not. For each QA-pair candidate, we concatenate the input question q with the QA-pair (q', a'), and feed it through ALBERT, and project the CLS representation to a logit score. The model produces a distribution over the K QA-pairs via softmax, and is trained to minimize the negative log-likelihood of the correct QA-pair. We obtain training data in the following manner: for a training QA-pair, we retrieve the top $2K$ QA-pairs from PAQ using RePAQ's retriever. If one of the retrieved QA-pairs has the correct answer, we treat it as a positive, and randomly sample K-1 of the incorrect retrieved questions as negatives. We train with K=10, and rerank 50 QA-pairs at test time. The reranker improves accuracy at the expense of speed. However, as QA-pairs consist of fewer tokens than passages, the reranker is still faster than retrieve-and-read models, even for architectures such as ALBERT-xxlarge.

We show that PAQ and RePAQ provide accurate ODQA predictions, at the level of relatively recent large-scale retrieve-and-read systems such as RAG (Lewis et al., 2020b) on NaturalQuestions (Kwiatkowski et al., 2019a) and TriviaQA (Joshi et al., 2017). PAQ instances are annotated with scores that reflect how likely we expect questions to appear, which can be used to control the memory footprint of RePAQ by pruning the KB accordingly. As a result, RePAQ becomes flexible, allowing us to configure QA systems with near state-of-the-art results, very small memory size, or inference speeds of over 1,000 questions per second.

PAQ can also be used as a source of training data for CBQA models. BART models trained on PAQ outperform standard data baselines by 5%. However, these models struggle to effectively memorise all the knowledge in PAQ, lagging behind RePAQ by 15%. This demonstrates the effectiveness of RePAQ at leveraging PAQ.

Finally, we show that as RePAQ's question matching score correlates well with QA accuracy, it effectively "knows when it doesn't know", allowing for *selective question answering* (Voorhees and Buckland, 2003) where systems may abstain from answering. Whilst answer abstaining is important in its own right, it also enables an elegant "back-off" approach where we can defer to a more accurate but expensive QA system when the answer confidence is low. This allows us to make use of the best of both speed and accuracy.

2.4 Parametric Models

The paradigms we previously explored in sections 2.2 and 2.3, namely the retriever-and-reader approach and the qa-pair retriever approach, can be classified as non-parametric models. This classification arises from their reliance on external resources to formulate answers to the provided questions. In contrast, the parametric model paradigm represents another distinct approach. Parametric models rely exclusively on the information stored within their parameters to formulate answers. These models encapsulate their understanding of the given knowledge solely through their learned parameters, making them self-contained in generating responses. They are also referred to as "closed-book" models. They are directly trained with QA pairs without access to an external corpus and thus store the required knowledge in its entirety in the model parameters. Previous work has analyzed the generative models such as BART-large and T5-11B (Lewis et al., 2021b). According to the previous observations and the experiments in Sec. 3, parametric models with more parameters are more effective at memorizing knowledge acquired at training time. However, they struggle to generalize to novel questions, with some model architectures showing no meaningful generalization capabilities at all. Additionally, there exists a large performance gap between non-parametric and parametric models. Liang et al. (2022) an extensive assessment of 30 prominent language models, scrutinizing their capabilities in 26 specific contexts. Among which, the evaluations on the closed-book NaturalQuestions (Kwiatkowski et al., 2019b) dataset show that InstructGPT davinci v2 (175B) (Brown et al., 2020) demonstrates superior performance for all knowledge-intensive evaluations. Further, TNLG v2 (530B) (Smith et al., 2022) shows strong performance on the NQ dataset, which generally concurs with the hypothesis that model scale especially contributes to improvements in acquisition of factual knowledge.

2.5 Evaluation

In assessing the effectiveness of the retriever, we utilize the standard *top-K accuracy* metric. This metric gauges the proportion of questions where, among the highest K retrieved passages, there exists at least one passage containing a sequence of words that aligns with the human-annotated answer(s) for the given question.

The final predicted answers are evaluated with the standard *exact match* matric, determining whether the system's generated answer exactly matches the human-annotated correct answer for a given question. If the generated answer is identical to the correct answer, it is considered a successful exact match. However, if there is any difference in wording, phrasing, or structure, the match is considered unsuccessful. This metric provides a stringent evaluation criterion, as even minor discrepancies between the system's answer and the correct answer lead to a failure in achieving an exact match. The exact match metric is valuable for gauging the ODQA system's ability to provide highly accurate responses that align precisely with the expected correct answers.

In comparison, the F1-Score is considered a less stringent metric than Exact

Match. This metric evaluates the mean overlap between the predicted response and the ground-truth answer. By treating both the prediction and the correct answer as collections of tokens, their F1-Score is calculated. For each question, the highest F1-Score across all correct answers is considered, and then this is averaged across all questions (Rajpurkar et al., 2016). While the F1-Score is generally a suitable measure for many span-based QA datasets question-answering datasets, its effectiveness can vary depending on the nature of the questions and answers. Particularly, in scenarios where incorrect and correct answers share common n-grams, the F1-Score may face challenges in accurately determining the quality of responses (Chen et al., 2019). This suggests a need for caution when applying this metric to different contexts or types of data.

Exact-match (EM) and F1-score, both lexical matching metrics, have a notable limitation in their inability to recognize certain valid answers. These metrics often fail to account for scenarios where a model's response, although not aligning verbatim with the expected answer, is still conceptually correct. This includes instances of Semantic Equivalence, where model predictions and the standard "gold" answers convey identical meanings but differ in their phrasing. Similarly, Symbolic Equivalence issues arise when numeric responses are correct but presented differently in text form. Moreover, Granularity Discrepancies are evident when the predicted answers differ in detail or scope compared to the gold answers. Additionally, these metrics struggle to navigate situations involving Incorrect Gold Answers, an issue stemming from data quality challenges (Kamalloo et al., 2023). These cases highlight the need for more nuanced evaluation methods that can appreciate the subtleties of language and meaning beyond mere lexical matching.

To augment lexical-based metrics, employing large language models (LLMs) as evaluators presents a promising alternative. In evaluation of question-answering models, Kamalloo et al., 2023 examined the capabilities of both GPT-4 (OpenAI, 2023) and Instruct-GPT (Ouyang et al., 2022). Their findings revealed that the evaluation results from GPT-4 were in line with those observed from InstructGPT, with slight enhancements. Notably, they observed a consistent average increase in accuracy for all models when assessed using GPT-4, a level of improvement comparable to that seen with InstructGPT. Additionally, in a similar trend to InstructGPT, the accuracy rates from GPT-4 evaluations were found to be marginally lower, around a small percentage, than those derived from human judgment. The study concludes that using zero-shot prompting in LLMs as an evaluation method could serve as a feasible alternative to human assessment, although it cannot detect unattributability in long-form answers.

Chapter 3

Challenges in Generalization in Open Domain Question Answering

Recent work on Open Domain Question Answering has shown that there is a large discrepancy in model performance between questions and answers observed at train time and *novel* questions and answers – even if they are derived from the same distribution (Lewis et al., 2021b). This raises the question: "What are the aspects of these novel questions that make generalization challenging?" which we seek to explore in this chapter.

In work on systematic generalization (Bahdanau et al., 2018; Lake and Baroni, 2018; Ruis et al., 2020), it is argued that even though a model has only observed a very small subset of all possible combinations of facts during training time, a good model should be able to generalize to all possible combinations of facts at test time. We draw upon these ideas to study generalization for ODQA and define the following three categories to support our investigation: *training set overlap*, *compositional generalization*, and *novel-entity generalization*. See Figure 3.1 for definitions and examples. Our categorization breakdown is motivated by how they capture different levels of generalization: *overlap* requiring no generalization beyond recognizing paraphrases, *comp-gen* requiring generalization to novel compositions of previously observed entities and structures, and *novel-entity* requiring generalization to entities not present in the training set. It is worth noting that we explicitly study in-distribution generalization rather than out-of-distribution generalization (such as

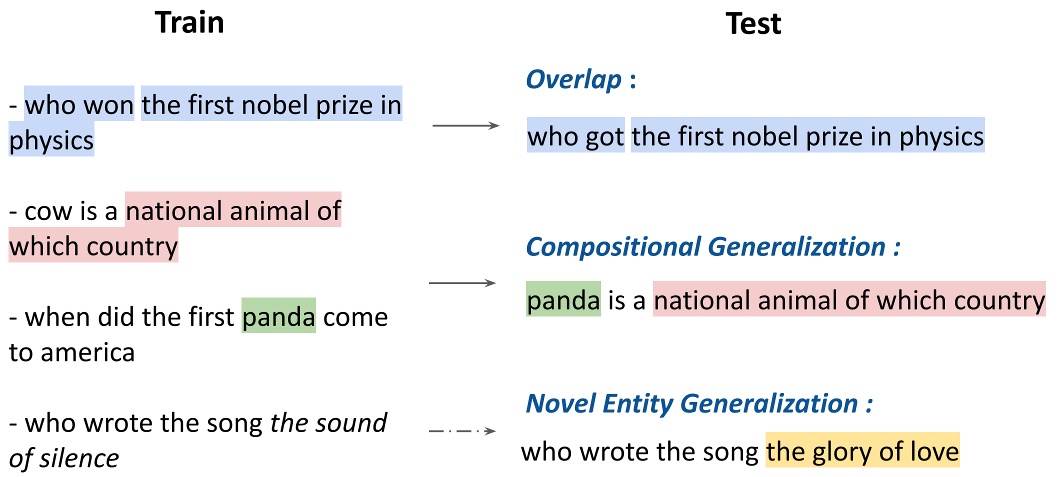

Train
Test
- who won the first nobel prize in physics
- cow is a national animal of which country
- when did the first panda come to america
- who wrote the song *the sound of silence*
Overlap :
who got the first nobel prize in physics
Compositional Generalization :
panda is a national animal of which country
Novel Entity Generalization :
who wrote the song the glory of love

Figure 3.1: Questions categorized according to their relation to the training set: 1) *Overlap*: there exists a paraphrase of the question in the training set. 2) *Compositional*: all individual facts and the structure of the question has been observed across several questions in the training set – but not the given composition. 3) *Novel-entity*: the question contains at least one entity (marked here with yellow) not present in the training set.

cross-domain generalization (Fisch et al., 2019)), as we will later demonstrate that even in-distribution generalization poses a major challenge for existing approaches.

We decompose and manually annotate three previously introduced ODQA datasets (Natural Questions (Lee et al., 2019), TriviaQA (Joshi et al., 2017), and WebQuestions (Berant et al., 2013)). Following this, we evaluate six recently proposed non-parametric and parametric ODQA models and analyze their performance, using both aggregate metrics and a breakdown according to our proposed categories. Non-parametric and parametric models differ in their access to information: the former has no access to any external context or knowledge, whereas the latter is provided relevant information alongside the question (Roberts et al., 2020). Experimental results show that the performance of non-parametric models degrades significantly on the comp-gen subsets across all datasets. We further examine what the underlying challenge is for these questions.

One potential source of difficulty could be the question structure itself and as a byproduct of our decomposition approach we are able to derive a high-level *question pattern* for each question. We find a strong positive correlation between the pattern frequency in the training set and test accuracy. We then study how non-parametric models handle the comp-gen and novel-entity subsets respectively, since the performance on them is significantly worse than on the overlap subset. For *comp-gen* questions, perhaps surprisingly, we find that the frequency of entities mentioned in a question is strongly *negatively* correlated with test accuracy. For *novel-entity* questions, when we replace novel entities in the question and its support passages with entities seen in the training set the performance remains largely unchanged; we thus hypothesize that specific unseen entities are not the main bottleneck for model performance but rather a failure of the model to generalise compositionally. Aside from questions, we further analyze the retrieved passages and find the retrieval accuracy is equally lacking for the *comp-gen* and *novel-entity* subsets, at $\sim 75\%$ for top-20 accuracy. We also observe that many of the passages that do contain the correct answer lack sufficiently informative contexts for the question anchor words for the reader model to be able to locate it, indicating a need to either improve the reader models ability reason over multiple passages or the retriever model to provide passages with richer contexts.

To conclude, in this chapter, our key contributions are as follows:

1. We provide the first detailed study on generalization for ODQA, based on categories that measure different levels and kinds of generalization, that we use to annotate three previously proposed ODQA datasets.

2. We show that for novel questions, non-parametric models handle novel question entities comparatively well, while they struggle to perform compositional generalization.

3. We demonstrate and quantify key factors that impact model generalization performance, which we believe will show the direction for future research towards more robust and generalizable ODQA models.

The material in this chapter first appeared in:

> **Linqing Liu**, Patrick Lewis, Sebastian Riedel, and Pontus Stenetorp.
> "Challenges in Generalization in Open Domain Question Answering."

In *Findings of the Association for Computational Linguistics: NAACL 2022, pp. 2014-2029. 2022.*

Individual Contributions: The initial idea was proposed by the thesis author. The dataset annotation pipeline and preparations, results analyses, experiments are conducted by the thesis author. The collaborative effort of all authors was integral in the human annotations of the three datasets.

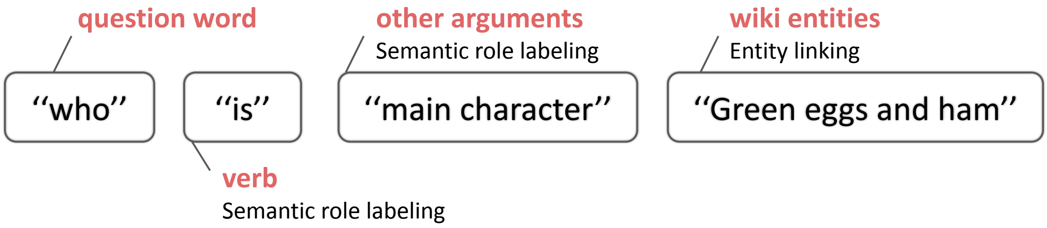


Figure 3.2: Example decomposition for the question *"Who is the main character in Green eggs and ham?"*

3.1 Dataset Construction

In this section, we describe how we process and annotate ODQA datasets to enable us to investigate generalization.

3.1.1 Question Decomposition

To study the compositional and novel-entity generalization of questions, we follow Keysers et al. (2019) and propose to view each question as being composed of primitive elements (atoms). Consider the question *"Who got the first Nobel Prize in Physics?"*. The atoms intuitively correspond to the modifier or adjunct of the predicate "who", predicate "got" and the entity "first nobel prize in physics". The combination of these atoms cover the main semantics of the question.

The way we measure generalization necessarily depends on how we break down the questions into atoms. Following manual analysis of questions from three popular ODQA datasets, we developed the following decomposition strategy to obtain atoms which cover all the desired question semantics. These are: question words, verbs, Wikipedia named entities (*wiki_entities*), and finally, other arguments (*other_args*) which correspond to other relevant aspects of the question. We explicitly extract wiki_entities since they leverage crucial semantics in factoid questions and other_args define essential details surrounding wiki_entities.

In order to automatically decompose questions, we first use an off-the-shelf semantic role labeling (SRL) model (Shi and Lin, 2019) to produce predicate-argument structures for each question. This provides us with the verb (i.e. the predicate), and semantic arguments. The question word is trivially obtained by identifying WH-words. We apply an off-the-shelf entity linking model (Li et al., 2020a) to obtain the wiki_entities in the question. Finally, other_args are the SRL arguments which remain after we filter out arguments corresponding to wiki_entities. An example question decomposition is illustrated in Figure 3.2.

Below is a random selection of question decomposition examples from the NQ dataset. In each question, $\underline{\text{x}}_{qw}$ denotes the question_word, $\underline{\text{y}}_{verb}$ denotes the verb, and the spans of other_args and wiki_ents spans are denoted by brackets. Note that these structure slots are not always fully present in the question (e.g, Q3, Q4, Q6, Q7, Q10).

As we rely on automated systems as a part of our decomposition process, this leads to the following limitations. At times, the ELQ model fails to label wiki_ents, such as for Q8 where *every light in the house* is marked as other_args. Furthermore, as seen in Q9 there is the possibility of multiple question words being present although our approach only extracts a single question_word. Limitations such as these is one motivation for why we elected to perform manual verification for each question (Section 3.1.3).

1. $\underline{\text{Who}}_{qw}$ $\underline{\text{is}}_{verb}$ the [*other_args*: owner] of [*wiki_entities*: Reading Football Club]?

2. $\underline{\text{Who}}_{qw}$ $\underline{\text{died}}_{verb}$ in the [*other_args*: plane crash] [*wiki_entities*: Grey's Anatomy]?

3. [*other_args*: Cast] of [*wiki_entities*: Law & Order Special Victim Unit]?

4. $\underline{\text{When}}_{qw}$ did [*wiki_entities*: United States] $\underline{\text{enter}}_{verb}$ [*wiki_entities*: World War I]?

5. $\underline{\text{Where}}_{qw}$ are most [*wiki_entities*: nutrients] $\underline{\text{absorbed}}_{verb}$ in the [*wiki_entities*: human digestive tract]?

6. $\underline{\text{When}}_{qw}$ did the [*other_args*: government] $\underline{\text{change}}_{verb}$ the [*other_args*: retirement age]?

7. $\underline{\text{What}}_{qw}$ $\underline{\text{is}}_{verb}$ the [*other_args*: name] of the [*other_args*: gap] between [*other_args*: two front teeth]?

Group	Test question	Paired training question for annotator	Label
Overlap	who got the first nobel prize in physics	who won the first nobel prize in physics	T
	whens the last time the patriots played the eagles	when did the philadelphia eagles last win the super bowl	F
Comp-gen	when is the next scandal episode coming out	when is next fairy tail episode coming out	T
	what is the corporate tax rate in great britain	what is the rate of corporation tax in uk	F
Novel-entity	who wrote the song the *glory of love*	who sang *guilty of love* in the first degree	T
	who sings *too much time on my hands* lyrics	who sings *i've got too much time on my hands*	F

Table 3.1: Example of questions from Natural Questions (see Appendix A.2 for examples from the other two datasets) for human verification and their respective annotated labels (T for True and F for False).

8. $\underline{\text{Who}}_{qw}$ $\underline{\text{sings}}_{verb}$ [*other_args*: every light in the house is on]?
9. $\underline{\text{Where}}_{qw}$ $\underline{\text{are}}_{verb}$ the [*wiki_entities*: Winter Olympics] and when do they start?
10. [*wiki_entities*: Swan Lake] [*wiki_entities*: the Sleeping Beauty] and [*wiki_entities*: the Nutcracker] $\underline{\text{are}}_{verb}$ [*other_args*: three famous ballets] by?

3.1.2 Generalization Category Definitions

Based on the question decomposition, we define three generalization categories for ODQA datasets. We denote S_q as the set of the decomposed atoms of question q and C_Q as the complete set of decomposed atoms for all the questions in dataset Q. Our category subsets are then defined as:

- $Q_{\text{overlap}} \triangleq \{q \in Q_{\text{test}} \mid \exists q' \in Q_{\text{train}}, S_q \subseteq S_{q'}\}$
- $Q_{\text{comp\_gen}} \triangleq \{q \in Q_{\text{test}} \mid \exists q'_1, q'_2, \ldots, q'_k \in Q_{\text{train}}, S_q \subseteq \bigcup_{i=1}^{k} S_{q'_i}, S_q \nsubseteq S_{q'_i}\}$
- $Q_{\text{novel\_entity}} \triangleq \{q \in Q_{\text{test}} \mid \exists s \in S_q, s \notin C_{\text{train}}\}$

For overlap test question, there exists a training question where they have the same decomposed atoms or are subset of them; for comp_gen test question, its decomposed atoms are fully covered by the training set (a subset of the union of multiple training questions atoms), but not in one particular training question; and for novel-entity test question, there exist wiki_entities not present in the training set.

3.1.3 Question Categorization and Human Verification

With the decomposed atoms for all questions, we first categorize the test questions into overlap, comp-gen, and novel-entity categories based on the definitions of each

Group	Natural Questions	WebQ	TriviaQA
Overlap	837	501	458
Comp-gen	1,105	512	475
Novel-entity	597	640	456

Table 3.2: Number of questions for each generalization subset for the three datasets' test sets

generalization category. We optimize the selection criteria to cover as many eligible candidates for each category as possible.

We use the following selection criteria to collect candidate questions for human verification. For the overlap subset, as a first step, each q is paired with each train question that shares the same answer or have answers which are a sub-sequence of q's answer. As a second step, we then require that the train question's similarity measurement score to q is over a pre-defined threshold and that they have the same wiki_entities as q. For the remaining test questions, we consider q as a candidate for comp-gen if all of its parsed elements are covered by the collection of all parsed elements in the training set. Lastly, if there exists any novel wiki_entities in q which are not present in the training set, q is considered as a novel-entity candidate.

As our test set subsets are obtained automatically, we need to perform manual human verification to ensure that they are of high enough quality to draw empirical conclusions. To do this, we employ four expert annotators and use the following annotation process for each of the respective categories. *Overlap:* Annotators are shown q_{test} and the training questions with the highest degree of character-level overlap. If any of these questions are a paraphrase of q, the annotator will mark q_{test} as an *overlap* question. *Comp-gen:* q_{test} is presented to the annotators along with the training questions with the highest degree of word overlap. Annotators then verify that the test question is truly a compositional generalization and not a paraphrase of any of the given training questions. *Novel-entity:* Annotators need to: 1) Verify that the wiki_entities identified by the entity-linking model are indeed wiki entities. 2) Verify that the entities in q_{test} are not present among a set of questions from the training set whose entities have a high degree of character-level overlap with the entities in q_{test}. Statistics for the annotated category subsets are summarized in Table 3.2, examples are shown in Table 3.1.

As guidelines for the human annotators, we provide the following to resolve ambiguous or potentially problematic cases: 1) For overlap, we only consider questions that are superficial paraphrases and exclude those that require more complex forms of reasoning (e.g. *Who played Mark on the show The Rifleman? / Who played the boy on the show The Rifleman?*). 2) For comp-gen, all other_args in the test question must be covered in the collection of training set entities and all question_word atoms alongside with the verb must be present in the training set. However, there are questions where other_args are not covered in the training set (e.g. *Animation Resort*) or are highly specific due to the decomposition processing and thus not covered (e.g. *fourth movie* compared to *movie* or *three different types* compared to *types*) and are thus excluded from comp-gen. 3) For novel-entity, there are cases when ELQ fails to extract wiki_ents in questions because of words variation, such as *Who sang It Going to Take a Miracle?* compared to the correct wiki_ents *It's Gonna Take a Miracle*. 4) There are also intrinsic problems in the datasets, some test questions are exactly the same as train questions but paired with different answers: (*Where did Dolly Parton grow up?* with the answer *Tennessee* and *Where did Dolly Parton grew up* with the answer *Sevierville*). Following this manual verification, for Natural Questions, WebQuestions, and TriviaQA, 70.3%, 81.3%, and 69.5% of their test questions are covered in the generalization subsets respectively.

3.2 Experiment

3.2.1 Datasets

We analyse three widely used ODQA datasets, each one is briefly introduced as follows:

Open Natural Questions (NQ) is an open-domain variant of Natural Questions (Kwiatkowski et al., 2019a) introduced by Lee et al. (2019). This dataset consists of questions mined from Google search logs, with answers annotated as short spans of text in Wikipedia articles by crowd-workers. The NQ questions are generally simple, short, and *information-seeking*, as the questioner is unlikely to have known the question's answer when they formulated it. It consists of 79,168 train, 8,757 dev, and 3,610 test question answer pairs.

TriviaQA (Joshi et al., 2017) consists of questions and answers which were obtained by scraping trivia websites. TriviaQA questions are generally less information-seeking than those in NQ, and exhibit substantial syntactic and lexical variability. We use the open domain splits which contains 78,785 train, 8,837 dev, and 11,313 test question answer pairs (Lee et al., 2019). Answers in TriviaQA are Wikipedia entities, and any alias of the answer entity is considered a correct answer. We randomly sampled and annotated 2,000 questions from the test set for our analyses.

WebQuestions (Berant et al., 2013) consists of questions that were collected by performing a breadth-first search using the Google Suggest API. The questions in WebQuestions resemble those in NQ, but are generally shorter and simpler and demonstrate less variability. WebQuestions' answers are Freebase (Bollacker et al., 2008) entities, annotated by crowdworkers. It contains 3,778 train and 2,032 test questions.

3.2.2 Baseline Models

Non-parametric models mostly adopt a retrieve-and-read framework, retrieving relevant Wikipedia documents for the given question, and then produce the final answer conditioned on these documents. We consider two generative reader models: Retrieval-Augmented Generation (RAG, Lewis et al., 2020b), and Fusion-In-Decoder (FiD, Izacard and Grave, 2021). RAG combines a DPR (Karpukhin et al., 2020) dense retriever with a BART (Lewis et al., 2020a) generator, which are jointly fine-tuned end to end. FiD is a pipeline approach which uses DPR to retrieve a set of documents, and the decoder attends over all encoded document representations to generate the final answer. As an extractive reader model we use the reader component from DPR (Karpukhin et al., 2020). It extracts answer span from the highest-scoring document ranked from a passage selection model. We also include RePAQ (Lewis et al., 2021c), a QA-pair retriever which does *not* follow the retrieve-and-read paradigm. It retrieves QA-pairs from PAQ, a large resource of

Model		Natural Questions				TriviaQA				WebQuestions			
		Total	Overlap	Comp -gen	Novel -entity	Total	Overlap	Comp -gen	Novel -entity	Total	Overlap	Comp -gen	Novel -entity
Non-parametric	RAG	44.49	75.75	30.41	37.69	56.83	87.12	47.58	47.81	45.52	80.64	33.40	31.88
	FiD	**53.13**	78.85	**40.00**	**47.74**	**67.69**	**90.39**	**58.10**	**66.23**	-	-	-	-
	DPR	41.27	71.33	25.88	33.84	57.91	82.31	46.11	58.99	42.42	73.45	31.05	31.25
	RePAQ	47.26	78.61	34.21	36.85	52.06	89.08	42.95	38.38	-	-	-	-
Parametric	T5-11B+SSM	36.59	**81.48**	17.47	12.56	-	-	-	-	44.69	81.24	35.35	25.78
	BART	26.54	76.34	5.88	3.35	26.78	78.38	11.37	10.09	27.41	70.46	13.28	8.75

Table 3.3: Exact Match scores for each model. "Total" refers to the overall performance on the full test set. "Overlap", "Comp-gen", and "Novel-entity" refers to the model performance on the respective subset.

65M automatically-generated QA-pairs, returning the answer of the most relevant QA-pair.

Parametric models are directly trained with QA pairs without access to an external corpus and thus store the required knowledge in its entirety in the model parameters. For our analyses, we include a BART-large model (Lewis et al., 2020a) and a more powerful T5-11B model (Roberts et al., 2020). They are both trained with questions as input and output question-answer pairs.

3.2.3 Model Category Analysis

Table 4.3 shows the Exact Match scores for models on our test set splits.

Non-parametric models on novel-entity questions For the non-parametric models, EM scores on *novel-entity* questions are relatively close to their overall total scores, with an average drop by 6.5% and 3.1% on NQ and TriviaQA respectively, with the exception of WebQuestions. The questions in WebQuestions only contain a single entity, which also tend to be high frequency entities. However, due to the very small size of the WebQuestions training set, many of these questions are considered to be in the novel-entity subset, despite containing relatively frequent entities, which, with a larger training set, would likely be classified as comp-gen questions, querying various relations regarding known entities.

Non-parametric models on comp-gen questions Surprisingly, the performance of all non-parametric models degrades significantly on the *comp-gen subset* (drop by 14.2% on NQ, 10.2% on TriviaQA and 11.7% on WebQuestions). This finding suggests that non-parametric models struggle to perform compositional generalization, whereas they handle novel question entities comparatively well. We investigate this finding in greater detail in Section 3.3.

Parametric models on novel-entity and comp-gen questions parametric model performance drops significantly on both comp-gen and novel-entity subsets, but they achieve relatively higher EM scores on comp-gen questions. This indicates that novel-entity questions are more challenging for parametric models. This makes intuitive sense, since, for entities not seen during training, parametric models will struggle to “know" enough about the entity to generate a correct answer. In such cases, we find evidence that parametric models often resort to generating answers from superficially similar training questions, with 63.2% and 53.3% of answer predictions also occurring in the training data for T5-11B+SSM on NQ for comp-gen and novel-entity questions respectively.

Implications for modeling Among the non-parametric models, FiD achieves the highest EM scores for both comp-gen and novel-entity questions. FiD aggregates multiple passages together when generating answers. In contrast, the extractive DPR reader only uses the highest-scoring passage to extract the final answer. We take a step further and are interested in understanding if FiD’s improved performance is due to leveraging a greater amount of contextual evidence provided by multiple passages, or whether it simply generates the most frequently-mentioned plausible answer. We perform a simple experiment, by first collecting 544 questions answered incorrectly by FiD, where the gold answers occur less frequently than FiD’s predicted answer in the retrieved passages. We then adjust the retrieved passages so that the original predicted answer and gold answer are mentioned an equal number of times, by masking out some of the original prediction mentions. After adjusting the frequencies, we regenerate the answer predictions, and observe that FiD only produces 44 correct answers out of 544. This suggests that answer mention frequency is not the governing feature for FiD when generating answers on NQ. It suggests the NQ FiD model adopts a strategy similar to a reranker, and extracts an answer from the highest latently-relevant document.

Although without access to external knowledge but only automatically-

NQ	Total	Overlap	Comp-gen	Novel-entity
Top-20	80.1	89.5	74.7	75.4
Top-100	86.1	92.0	82.4	83.1

Table 3.4: Top 20 and Top 100 retrieval accuracy on NQ test set for the DPR retriever.

generated QA-pairs in advance when answering questions, RePAQ still achieves higher or comparable performance as retrieve-and-read model RAG and DPR. It indicates that generating, storing and retrieving questions is a valid path in terms of model generalization.

Parametric models perform significantly worse compared to non-parametric models. BART struggles to answer any novel questions correctly, while T5-11B+SSM performs better due to much larger capacity. Petroni et al. (2019) demonstrate that language models are able to recall factual knowledge without any fine-tuning and can somewhat function as an unsupervised ODQA system. However, our experiments suggest that, large-scale language models (when fine-tuned to directly answer questions using a set of training QA pairs) struggle to answer questions about low frequency entities and relations, similar to the findings of Kassner et al. (2020) and Dufter et al. (2021).

Additional observations All models perform significantly higher on overlap questions, consistent with the findings of Lewis et al. (2021b). Parametric models with more parameters are the most effective at rote-memorizing training questions, and T5-11B+SSM even outperforms the non-parametric models on NQ and WebQuestions.

3.3 How do Non-parametric Models Generalize?

Experimental results show that the performance of non-parametric models degrades significantly on the comp-gen subsets across all datasets. In this section, we would like to examine what the underlying challenge is for these questions. We focus on the NQ dataset as it has the largest annotated test set among three datasets.

Table 3.4 shows the top-*k* retrieval accuracy – which is the number of questions

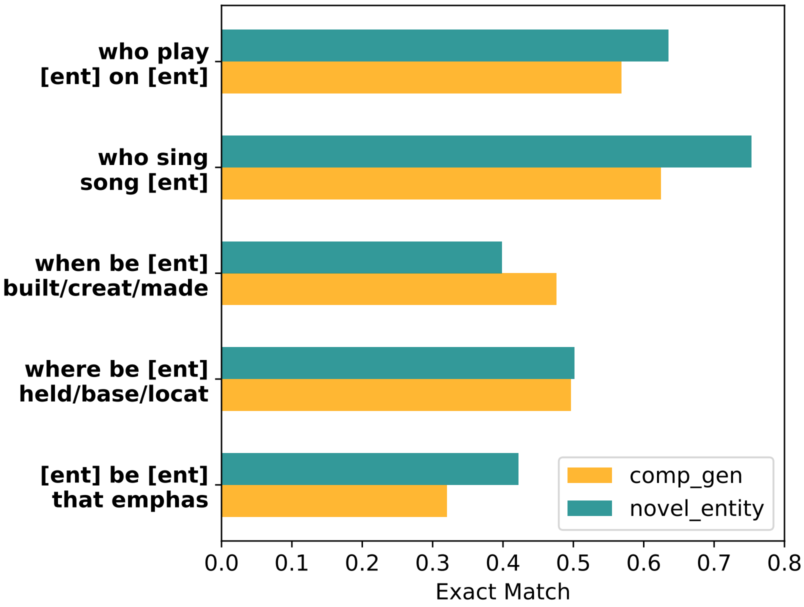


Figure 3.3: Examples of question patterns and EM scores for their corresponding questions. For each question pattern, we sample the same number of comp-gen and novel-entity questions. The two uppermost patterns are the most frequent (thousands of occurrences), the following two are of medium frequency (hundreds of occurrences), and the last is a novel pattern.

for which at least one passage of the top-k retrieved passages contains the gold answer. The difference in retrieval accuracy between comp-gen and novel-entity splits is relatively small ($< 1\%$), but is significantly lower than the overlap subset results. This indicates that the retriever performance is a confounding factor for the overall performance of comp-gen and novel-entity questions. Solely improving the retriever would benefit the model greatly for the subsets requiring generalization. Allowing us to study the reader model in isolation, for the remainder of our analysis we will only use the subset of questions for which there is at least one support passage that contains the gold answer.

3.3.1 Effects of Question Pattern Frequency

One might ask questions such as *"Who plays the doctor in Sons of Anarchy?"* and *"Who plays Stacey's mum in Gavin and Stacey?"*. Although semantically different, they share the structure "who plays [entity] in [entity]", which we refer to as a question pattern. To study if the frequency of these patterns affect model performance,

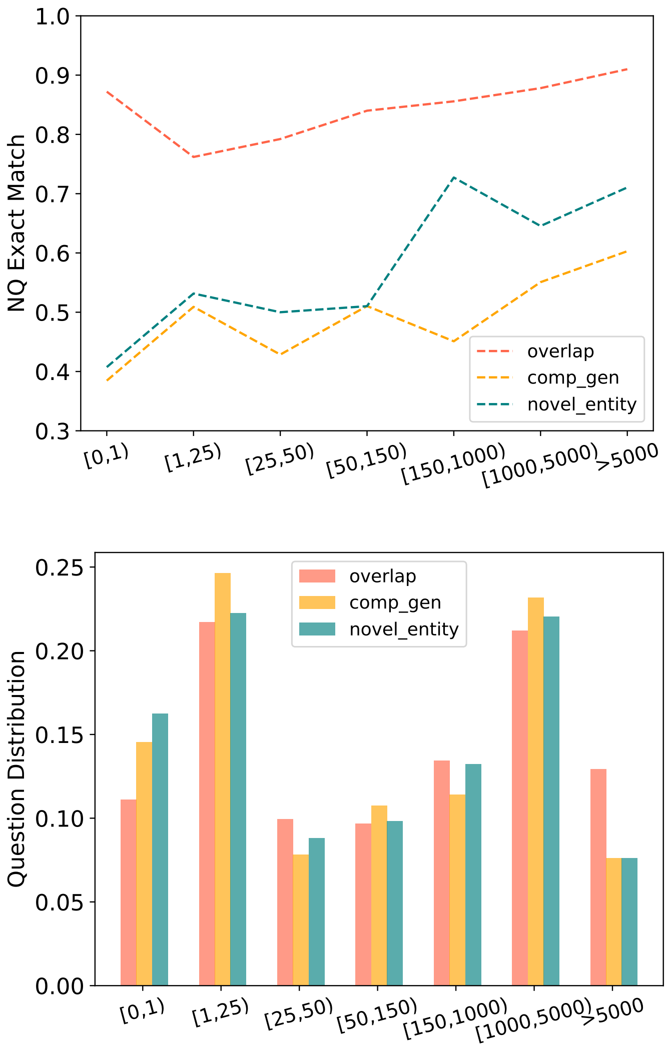


Figure 3.4: Influence of question pattern frequency, where test questions are binned based on the frequency of their question pattern in the *training set*.

we collect question patterns by replacing all wiki_entities in a question with the token *[entity]*, unifying the prepositions, and stemming each word.

We group test questions for each category by the frequency of their patterns in the training set. In Figure 3.4, we analyze FiD as an example since it achieves the highest EM score on unseen questions (results for other models can be found in Figure A.1 in the Appendix). In the upper figure, the EM scores show that the model is more likely to make correct predictions for more common patterns. Given this observation, we would like to investigate if the significant performance edge of the overlap category is due to a larger percentage of more frequent patterns. According to the lower figure, which shows the proportion of questions for each frequency bin, the frequency distribution for each category is largely similar. Therefore the performance gap between overlap and the other two categories can not simply be explained by a difference in pattern distribution.

In Figure 3.4, we also note that as the pattern frequency increases, the performance between comp-gen and novel-entity diverges. This gap has a significant effect on overall model performance, since common patterns make up a majority of the test set. In order to understand this gap, we sample the same number of comp-gen and novel-entity questions for each example pattern, and display the results in Figure 3.3. We checked several instances for the pattern "who play [ent] on [ent]", and find that the model fails more on comp-gen questions partially because the retrieved passages do not provide enough information to locate the answer. For example, for the question *"Who played Mary in Christmas with the Kranks?"* none of the retrieved passages contain both *Mary* and the movie name. The model produces the answer *Julie Gonzalo* from the passage *Julie Gonzalo Julieta [...] is an [...] actress. [She] is also known for her roles "Christmas with the Kranks"*, whereas the gold answer is *Felicity Huffman* from the passage *She also starred in [...] "Christmas with the Kranks"*. Since "Mary" is not mentioned in either passage, it is impossible to infer that the correct answer is *Felicity Huffman*. The support passages for novel-entity questions, on the contrary, more often cover both of the anchor entities (e.g. context *Little Boy Blue is an ITV drama series ... Stephen Graham was cast as Detective ...* for the question *"Who played the detective in Little Boy Blue*").

Based on the above error analysis, we hypothesise that in the retrieved passages for comp-gen questions, answers do not always co-locate with the question anchor words. This indicates future research should encourage the retriever to fetch passages that cover all aspects of the question in order for it to be answerable. Under the assumption that the model could answer all patterns of questions equally well, regardless of frequency, the overall performance would be improved by $\sim 11\%$.

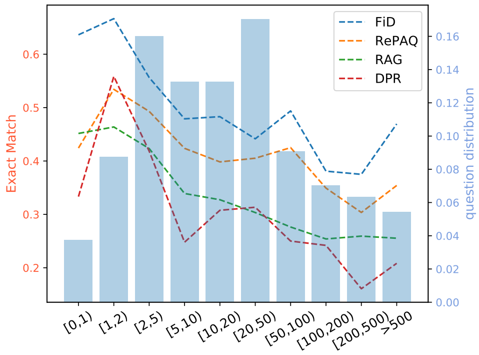


Figure 3.5: Plot showing the influence of the wiki_entities frequency in the question. The x-axis represents the wiki_entities frequency in the training set and we use the most frequent wiki_entities in each comp_gen question.

3.3.2 How do Non-parametric Models Handle Comp-gen Questions?

We use the decomposed atoms as the basis for our analyses on comp-gen questions. Following the previous subsection 3.1.1, we know that wiki_entities leverage crucial semantics for factoid questions and Wikipedia is the most widely used source of knowledge in current ODQA datasets (Hewlett et al., 2016; Rogers et al., 2021; Yang et al., 2015). Therefore, we would like to carefully study if the training set **wiki_entities** frequency affects model performance. Figure 3.5 plots the EM score as a function of how often a test question's wiki entity appears in a training question. We see that test accuracy is *anti-correlated* with the training-set frequency of test questions' entities. At first glance, this result seems surprising, and inconsistent with the well-known difficulty of modeling long-tail phenomena. However, the following interpretation helps to explain this apparent contradiction.

We manually inspect the questions with the most frequent wiki_entities, and find most of them are questions about countries, which is a frequent question topic in the NQ training set. For example, for the question *"How many farmers are there in the USA"*, almost all the retrieved passages are highly relevant. The gold answer is "3.2 million" with the context *"There were 3.2 million farmers"*. The model, however, generates the answer "2.2 million", taken from the context *"There were 2.2 million farms…"*. Both passages come from an article titled "Agriculture in the United States", and the model is failing to draw a distinction between *farms* and *farmers*. While it is easier to retrieve relevant documents for questions with more frequent wiki_entities (Chen et al., 2021a), the passages retrieved for high-frequency entities are much more likely to contain type-consistent close-negatives and distractors, making it more difficult for the model to select the correct answer. In other cases, questions are highly ambiguous, such as, "*What is the average salary for a US congressman*", the gold answer *$174,000* applies for the year 2012, while predicted answer *$169,300* applies for the year 2008. For NQ, the existence of high-frequency entities could be indicative of an ambiguous question. If we conduct an analysis using the NQ dev set annotations provided by Min et al. (2020), we note that 50% of questions with the entity "*US*" and 64% of questions with the entity "*NBA*" are ambiguous. To quantify the impact, using FiD as an example, we note that if we match the performance of comp-gen questions with common wiki_entities to those with the unpopular wiki_entities, the accuracy could be improved with $\sim 4\%$ points.

Besides wiki_entities, it's prudent to consider the remaining atoms as well. The results are illustrated in Figure 3.6 and some findings are observed in the following: 1) For **question word**, all models achieve better performance for questions asking about WHO and WHICH, while performing worse on questions without any question word. Although EM scores drop significantly for WHY questions, it is hard to draw conclusions as there are only limited number of them in the test set. 2) There is no clear correlation between model performance and **verb** frequency. Some of the "best performing" verbs are: sing, sang, wrote, and play, which closely correlate with the most frequent question patterns such as "who sing song [ent]". 3) Since there is no clear correlation between model performance and other_args frequency either, we group test questions based on the number of **other_args** in each of the questions. It shows that models achieve higher EM scores on questions with fewer other_args. Interestingly, the most performing other_args are closely related to WHO and WHICH questions, such as "'s wife", "main character", and "tv show", while the "worse performed" other_args are mostly the comparative and superlative adjectives

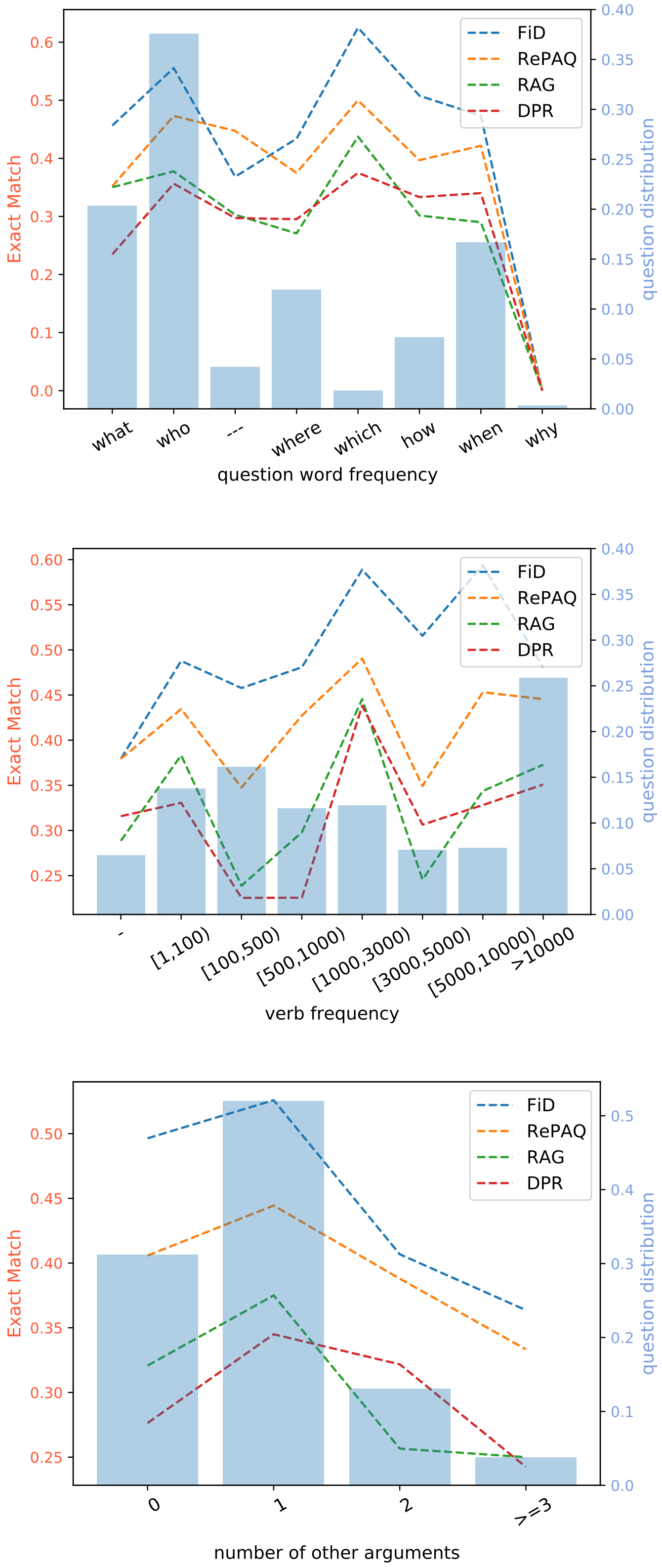


Figure 3.6: Influence of *question word*, *verb*, and *other_args* in the question (from left to right). In the two top figures, the test questions are binned based on the individual atom frequency in the training set, "-" indicates test questions whose question word or verb is not covered in the training set. In the bottom figure, the x-axis shows the number of other_args in each test question. All models are evaluated on the NQ test set.

such as “biggest house” and “second largest” (also observed in Dua et al., 2019).

To summarize, the *remaining atoms* are codependent on each other, especially for limited-length factoid questions. They should preferably be treated as a single unit (e.g. question pattern) to arrive the meaning of the question. In essence, their compositionality cannot be ensured and isolated (Dankers et al., 2021). Wiki_entities on the other hands are independent of the context. The question is meaning-preserving even under wiki_entities substitution. The subpart for ODQA compositionality should focus on wiki_entities and question patterns. As discussed above, their individual frequency have different impacts on the various components of ODQA models.

3.3.3 How do Non-parametric Models Handle Novel-entity Questions?

Although we explicitly categorize unseen questions into comp-gen and novel-entity, broadly speaking, questions with novel entities also require the model to generalize to novel compositions and thus could be considered to belong to the comp-gen category. We seek to understand if the novel entities are the main bottleneck for ODQA models, or the model can handle them well enough to process the questions appropriately. To explore this issue further, we run an ablation study, where, at inference time, we replace the novel entities in the question *and* the support passages with an entity that has been seen from the training set. Our experimental setup is working under the following constraints: 1) There can be only one wiki_entity mentioned in the test question, so that replacing it will not risk altering the semantics of the original question. 2) The replacement entity must not be present in the original test question or its retrieved passages.

We run the inference for FiD model on 100 eligible questions, and find the model rarely changes its predicted answers, despite the modification, with 73% of the predicted answers remaining unchanged. We manually verified the remaining questions and observe that some differences are due to inherent limitations of our entity-swapping process, such as errors in entity-linking. For instance, for the question *Who sings So Come and Dance with Me Jai Ho?* we swap the entity span “So Come and Dance with Me Jai Ho”, however, this span is too wide as an entity as the correct entity would be “Jai Ho”. Therefore the model is unable to match the correct song name in the passage; thus giving a different answer. Interestingly, we find that three altered questions give the right answers, despite originally generating incorrect ones. Given these observations, we suggest that the model learns relatively good contextual embeddings for the novel entities by exploiting the context provided by the passages. Thus, specific unseen entities are not the main bottleneck for the model to locate the desired answers.

3.4 Related Work

Retrieving relevant passages is an essential component for open-book ODQA models. A broad spectrum of recent work apply transformer (Vaswani et al., 2017b) models such as BERT (Devlin et al., 2019) for information retrieval (Yates et al., 2021). Following the success of using pretrained language models (Craswell et al., 2020), studies have been made regarding their properties. Luan et al. (2021) compare the lexical-matching abilities of these models to traditional methods such as BM25. Ma et al. (2021b) and Wang et al. (2021) study reproducibility, and demonstrate improvements by combining lexical-matching and dense retrievers. Thakur et al. (2021) introduce the BEIR benchmark to study zero-shot generalization for multiple neural retrieval approaches. Their conclusion is consistent with our findings that there is considerable room for improving the generalization of dense-retrieval models.

To infer answers from retrieved documents, models generally use a *reader* component implemented as a neural Machine Reading Comprehension (MRC) model. Previous work has analyzed the MRC model by crafting adversarial attacks (Jia and Liang, 2017; Mudrakarta et al., 2018), studying the difficulty of popular benchmarks (Kaushik and Lipton, 2018), and demonstrating annotation bias (Chen and Durrett, 2019; Gururangan et al., 2018; Sugawara et al., 2018). Despite the success for various datasets, there is little work analyzing the whole pipeline of question answering systems. Lewis et al. (2021b) showed that current ODQA models competently memorize their training question answer pairs, but struggle to generalize to novel questions, with some model architectures showing no meaningful generalization capabilities at all.

Krishna et al. (2021) found that long-form question answering (LFQA) systems do not ground their answers in the retrieved passages, but rather generate the same answer regardless of which retrieved passages it is presented with for a given question. In contrast, for ODQA, we observe that when we replace retrieved passages with randomly-sampled passages at inference time, the model FiD (Izacard and Grave, 2021) largely fails to correctly answer any questions (see Appendix A.1 for experimental details). Gu et al. (2021) define similar generalization levels based on schemas for Knowledge Base Question Answering. However, our setting works without a schema and our generalization categories are derived from question decomposition atoms.

3.5 Conclusion

We study ODQA model generalization and categorize unseen questions into three subsets: *overlap*, *comp-gen*, *novel-entity*. Treating questions as being compositional, we decompose them into atomic elements based on their semantics. We believe that this decomposition strategy can help future work related to question structure and unification. We evaluated several recent ODQA models on these three subsets for three popular datasets. Our experimental findings both pinpoint the specific problems when handling different categories of novel questions and shed light on how to compositionally approach the factoid questions in ODQA task.

Chapter 4

Query Expansion Using Contextual Clue Sampling with Language Models

Despite the advent of dense retrieval approaches based on semantic matching for open-domain question answering such as DPR (Karpukhin et al., 2020), approaches based on lexical matching (e.g., BM25) remain important due to their space-efficiency and can serve as input to hybrid methods (Formal et al., 2021b; Gao et al., 2021; Lin and Ma, 2021b). A core challenge for lexical retrieval is the vocabulary mismatch between the query and documents. Query expansion techniques dating back over half a century have proven effective in overcoming this issue (Salton, 1971). The expansion terms are traditionally precomputed from relevant corpora using pseudo-relevance feedback techniques (Abdul-Jaleel et al., 2004; Robertson and Jones, 1976; Salton, 1971). In recent work, GAR (Mao et al., 2021) explored removing the query expansion's reliance on an external corpus and instead used a large language model to generate a context.

We argue that expansion needs to balance two key factors: (1) Diversity: Given the question, there can be multiple different reasoning paths (referred to as *contextual clues*) to reach the correct answer. (2) Relevance: Simply relying on a single generated context increases the risk of query drift, as the generated context could be semantically irrelevant or contain factual errors (Schütze et al., 2008). However, simply generating multiple contexts is prone to the *hallucination* problem – they can be unfaithful to the input or include false information (Dziri et al., 2021; Maynez

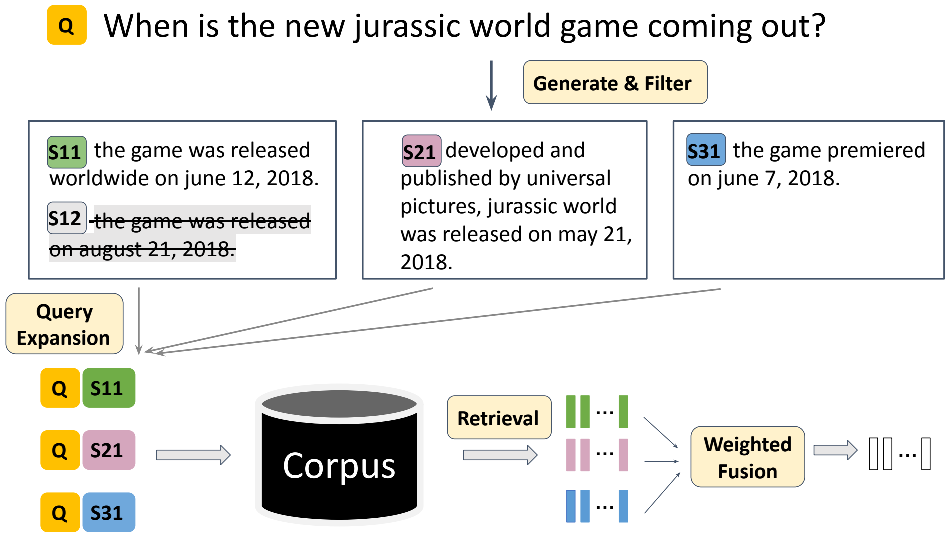

Q
When is the new jurassic world game coming out?
Generate & Filter
S11 the game was released worldwide on june 12, 2018.
S12 ~~the game was released on august 21, 2018.~~
S21 developed and published by universal pictures, jurassic world was released on may 21, 2018.
S31 the game premiered on june 7, 2018.
Query Expansion
Q S11
Q S21
Q S31
Corpus
Retrieval
Weighted Fusion

Figure 4.1: Given the query, we first sample various contextual clues from the language model. These outputs are then grouped together based on their lexical similarity and the one with the highest generation probability is retained (*Filtering*). We then perform retrieval for each augmented query individually. All the retrieved documents are ranked together in the final *fusion* step.

et al., 2020; Tian et al., 2019). Thus, in this work, we wish to explore the question: How can we best generate a sufficiently rich set of contextual clues to answer a query?

Our proposed solution (Figure 4.1) overcomes these problems with two simple and efficient steps: Filtering and fusion. After sampling top-k outputs from the decoder of the fine-tuned language model, we cluster these generated contextual clues based on their lexical distance. In each cluster, where highly similar contextual clues are grouped together, only the single output with the highest generation probability is kept. This *filtering* step effectively reduces potential factual errors and eliminates redundant close duplicates. Next, the query is individually augmented with each filtered contextual clue. We then retrieve documents separately for every single augmented query. Finally, as the last step, all the documents are ranked together (*fusion*) with the generation probability from the integral contextual clue in the augmented query.

We evaluate our approach on two established benchmarks: Natural Questions (Kwiatkowski et al., 2019a) and TriviaQA (Lee et al., 2019). Our base-

line model GAR (Mao et al., 2021) trails behind its dense retrieval counterpart DPR (Karpukhin et al., 2020) by a large margin when retrieving a small number of passages. We bridge this gap and outperform GAR by 3.1% and 2.9% on Top-5/Top-20 accuracy on the NQ dataset. Additionally, compared with DPR, our approach outperforms it by 0.6 and 1.0 points on Top-100 accuracy on the two datasets, while also requiring 96% less index storage space. This accuracy can be further improved by 3.4% when fusing the documents retrieved from DPR and our method together. Furthermore, our retrieval performance also successfully transfers to downstream question answering tasks, where our methods increase by 3.2% and 0.8% Exact Match score compared to the DPR and GAR retrieved documents.
The material in this chapter first appeared in:

> **Linqing Liu**, Minghan Li, Jimmy Lin, Sebastian Riedel, and Pontus Stenetorp. "Query Expansion Using Contextual Clue Sampling with Language Models." *arXiv preprint arXiv:2210.07093 (2022).*

> *Individual Contributions: The original idea was proposed by the thesis author. In the retrieval pipeline, the thesis author works on the contextual clue sampling and filtering, while the second author is responsible for document retrieval and fusion. The majority of the experiments and result analyses are performed by the thesis author. The major portion of the experiments and result analyses is conducted by the thesis author. In terms of paper composition, the second author primarily concentrates on introducing the retrieval and fusion methods, with the remaining sections authored by the thesis author.*

4.1 Background and Related Work

The task of information retrieval is to retrieve an ordered list of documents from a large corpus in order to respond to a specific query. As apposed to the neural network models where represent each query and document with learnt vectors, traditional approach uses exact term matching, comparing the exact words in the query to those in the document. With techniques based on exact term matching, the relationship between a query q and a document d can be scored as (Lin et al., 2022):

$$S(q,d) = \sum_{t \in q \cap d} f(t) \tag{4.1}$$

where f is a certain weight scheme and statistics on the terms. One of the most important weighting scheme is term frequency-inverse document frequency (tf-idf; Salton et al., 1975). *tf* calculates how many times a term occurs in a document, and *idf* is calculated by taking the logarithm of the ratio of the total number of documents in the corpus to the number of documents containing at least one instance of the term. Once both TF and IDF are calculated, the TF-IDF score for a term in a document is obtained by multiplying its TF and IDF values. The higher the TF-IDF score for a term in a document, the more important and relevant that term is to the document compared to other documents in the corpus. Terms that appear frequently in a specific document but are rare in the overall corpus tend to have higher TF-IDF scores, indicating their significance in that particular document.

After the proposal of TF-IDF, extensive research has been conducted to explore various term weighting schemes in the vector space model. Among these approaches, BM25 (Crestani et al., 1998; Robertson and Zaragoza, 2009; Robertson et al.) emerged as a prominent and enduring method used in numerous text ranking applications to this day. The calculation of BM25 is as follows:

$$\mathrm{BM25}(q,d) = \sum_{t \in q \cap d} \log \frac{N - \mathrm{df}(t) + 0.5}{\mathrm{df}(t) + 0.5} \times \frac{\mathrm{tf}(t,d) \times (k_1 + 1)}{\mathrm{tf}(t,d) + k_1 \times \left(1 - b + b \times \frac{\mathrm{len}(d)}{\mathrm{avg\_len}}\right)} \tag{4.2}$$

In the above equation, the left component is the *idf*, where N is the total number of documents in the corpus, and $\mathrm{df}(t)$ is the number of documents that contains the term t. In the right component of the equation, the expression in the denominator, which includes b, serves the purpose of conducting length normalization. This is necessary because collections typically consist of documents with varying lengths: $\mathrm{len}(d)$ stands for the length of document, while avg_len represents the average document length across all documents in the corpus. Addtionally, k_1 and b are tunable parameters in controlling the impact of term frequency and document length normalization, respectively.

BM25 offers several strengths, including its simplicity, ease of implementation, and robust retrieval performance across diverse domains and datasets. However, it is limited by the reliance on exact matching, which can lead to the "vocabulary mismatch problem" when queries and documents use different terminology. This becomes particularly pronounced in natural language queries and documents, where various words or phrases may convey the same meaning. For instance, if a user searches for "automobile," and a relevant document uses the term "car," BM25's exact matching approach may not recognize their semantic similarity, potentially resulting in a lower ranking score or even excluding the document from top results. Furthermore, BM25's lack of semantic understanding can make it sensitive to variations in linguistic phenomena, such as synonymy and paraphrase. Consequently, slight differences in wording between the query and the document may lead to reduced relevance scores, even when the essential content matches perfectly.

There are two directions of approach to tackle the above mentioned challenge for exact matching based retrieval: expand query to better match with the document representations, and enrich document to better match query representations. For the first direction, it is worth noting that query expansion techniques can be classified as pre-retrieval or post-retrieval techniques based on whether expansion terms are computed before or after examining documents from the collection (Lin et al., 2022). Relevance feedback (Robertson and Jones, 1976; Salton, 1971) and pseudo-relevance feedback (Croft and Harper, 1979) aim to improve the retrieval performance by incorporating user feedback or simulated feedback based on the assumption of relevance in the retrieval process. Other work also try to augment the query with lexical-semantic relations from WordNet (Miller, 1995) and preprocess the corpus to identify word relations as possible expansion terms (Xu and Croft, 2000). The expanded terms can also come from Domain knowledge source (Bouchoucha et al., 2013; Xiong and Callan, 2015), or selected from a Reinforcement Learning framework to maximiz the document recall reward (Nogueira and Cho, 2017). The second direction, document expansion, has been proven effective for noisy transcriptions of speech (Singhal and Pereira, 1999) and short texts such as tweets (Efron et al., 2012). Brauen et al., 1968 dynamically update the document representation after each retrieval run. Kwok, 1975 uses the citation metadata to enrich each document. Nogueira et al., 2019b firstly successfully apply neural networks to document expansion, naming their approach doc2query. They train a sequence-to-sequence model to produce possible relevant queries given an input document. In this work, we focus on improving the approach of query augmentation to enhance exact matching based information retrieval.

Methods	Natural Questions			TriviaQA		
	Top-5	Top-20	Top-100	Top-5	Top-20	Top-100
Ours-single (unfiltered)	61.1	73.7	84.1	70.9	78.7	84.3
Ours-single	63.0	75.2	84.8	71.7	79.1	84.6

Table 4.1: Top-5/20/100 retrieval accuracy (%) on Natural Questions and TriviaQA test sets. Filtering strategy effectively increases the retrieval accuracy and reduces the search space for the retrieval fusion step.

4.2 Methods

4.2.1 Contextual Clue Sampling and Filtering

We employ a sequence-to-sequence model BART-large (Lewis et al., 2020a) as our generator that takes the question as input and generates the contextual clues for the answer as target. It is worth noting that the generator can be replaced with any other sequence-to-sequence models. Contextual clues are the sentences in a passage that contains the ground-truth answer to the question. These sentences are either extracted from the passage provided by the dataset (when available), or from the matching passage used as a reference for the retriever.

At inference time, we first sample a diverse set of contextual clues from the fine-tuned model. Generally speaking, a single contextual clue can be broken into two main components; relational facts (“august 21, 2018”) and contextual description (“the game was released on”). Interestingly, we notice that many generations are identical in contextual descriptions, but inconsistent with the fact words (various dates, numbers or named entities). Previous works try to solve this inconsistency issue with an additional training loss (Elazar et al., 2021), adding a reasoning module (Nye et al., 2021), or through majority vote (Wang et al., 2022). Instead, we first cluster the contextual clues based on their edit distance. In most cases, the generated outputs with the same contextual descriptions but varying relational facts are grouped together in the same cluster. Then, we employ a simple filtering strategy for each cluster by keeping the top-ranked output with maximum generation probability, while discarding the rest outputs in the cluster. This allows us to gather all possible reasoning paths to the answer, while reducing potential factual errors.

Directly augmenting question with the full set of sampled contextual clues is a sub-optimal solution due to the following reasons: 1) Retrieval efficiency: After filtering, for each query, we only need to perform 70% less times of the retrieval. As a result, it's also saves the search space for the retrieval fusion step. 2) Retrieval accuracy: As shown in Table 4.1, the accuracy for the unfiltered contexts is consistently lower than that on the filtered contexts. We suppose it's due to removal of hallucinated facts contained in the contextual clues during filtering. Therefore this filtering strategy is crucial for the following retrieval step in terms of both retrieval *efficiency* (saves 70% for the retrieval process) and *accuracy* (consistently better than using full contextual clues).

4.2.2 Document Retrieval and Fusion

Defining the n generated and filtered contextual clues as $\{c_i\}_{i=1}^n$, we augment the question q into $\{\texttt{[CLS]}\,q\ \texttt{[SEP]}\ c_i\}_{i=1}^n$ by appending each individual context to it. Following GAR, we use BM25 as the retrieval backend where it could be considered as a logical scoring model using a query encoder η_q and passage encoder η_d (Lin, 2022):

$$s(q,d) = \phi(\eta_q(q), \eta_d(d)) \tag{4.3}$$

In this equation, ϕ is a similarity function such as dot product or L2 distance. Note that we use c to denote the generated contexts and d to denote real passages in the corpus. To aggregate the retrieval results of different augmented queries, we perform retrieval individually for each augmented query and use the likelihood $p(c_i \mid q)$ of the generated context c_i as the fusion weights. Therefore, the final retrieval score $s_f(q,d)$ for each question-passage pair is calculated as:

$$s_f(q,d) = \sum_{i=1}^{n} p(c_i \mid q) \cdot s(\texttt{[CLS]}\,q\ \texttt{[SEP]}\ c_i, d) \tag{4.4}$$

We finally re-sort the candidates according to the fusion scores and return the top-k passages for the next stage of question answering.

4.3 Experiment

4.3.1 Datasets

We conduct the experiments on two widely used ODQA datasets: Natural Questions (NQ) (Kwiatkowski et al., 2019a) and TriviaQA (Joshi et al., 2017). NQ consists of 79,168 train, 8,757 dev, and 3,610 test question-answer pairs. We use the open-domain splits of TriviaQA which contains 78,785 train, 8,837 dev, and 11,313 test QA pairs (Lee et al., 2019).

4.3.2 Experiment Setup

We finetune BART-large model (Lewis et al., 2020a) for contextual clue generation. For Natural Questions dataset, we extract the sentence containing the ground-truth answer from the provided positive passage. For TriviaQA dataset, since only pairs of questions and answers are provided in the original dataset, we extract the answer context sentence from the highest ranked passage retrieved by BM25. We train the model using Adam optimizer (Kingma and Ba, 2015b) with a learning rate of $3e-5$, linear scheduling with warm-up rate 0.01, and training batch size of 256 on 4 V100 GPUs.

Given the question, we generate 100 candidate outputs from BART using beam search with beam size 100. We first group similar candidates using fuzzy string matching with the built-in *difflib* python module. The similarity cutoff is set to 0.8 and any string pairs scoring less than the cutoff are not kept in the same group. On average, for each question there are 24 contextual clues for NQ and 33 for TriviaQA after filtering.

For each contextual clue augmented query, we use the Pyserini (Lin et al., 2021) BM25 to retrieve top-1000 candidate passages. All the retrieved documents are then re-ranked according to Eq. equation 4.4. We put all the passages belonging to the same question but different augmentation into a public pool after filtering duplicates. We then average the retrieval score for each passage in the pool according to Eq. equation 4.4 and re-sort the order of the fused passages. For fair comparison with GAR (Mao et al., 2021), we additionally fine-tune an answer generation model

# Context	ROUGE-1	ROUGE-2	ROUGE-L	Ans Cover
Top-1	35.27	22.82	31.84	29.02
Full	48.32	32.43	42.64	46.01
Filtered	47.14	31.44	41.80	43.17

Table 4.2: Evaluation of generated answer contexts on the validation set of the NQ dataset.

and a title generation model. We perform the same fusion steps above for all three generation models, and we linearly interpolate their fusion results by searching the best weighting on the development set using Bayesian Optimization (Frazier, 2018).

4.3.3 Baselines

Retriever Retrieval in open-domain QA is traditionally implemented with sparse vector space model BM25 (Robertson and Zaragoza, 2009), based on exact term matching. DPR (Karpukhin et al., 2020) implements retrieval by representing questions and passages as dense vectors. GAR (Mao et al., 2021) proposes to expand the query by adding relevant answers, the title of a passage and the sentence where the answer belongs. It also fuse the results from its own and from DPR (GAR+DPR). To make a fair comparison, we extend our generation target from the answer context only (*Ours-single*) to include both the answer and the passage title (*Ours-multi*). We also report the fusion results with DPR. SEAL (Bevilacqua et al., 2022) use BART model to generate ngrams then map to full passage with FM index.

Reader DPR (Karpukhin et al., 2020) employs a BERT-based (Devlin et al., 2019) extractive reader model and predicts the answer span. RAG (Lewis et al., 2020b) combines the DPR dense retriever together with a BART answer generator, and jointly trains the two models end-to-end. FiD (Izacard and Grave, 2021) also uses DPR to retrieve relevant passages and the decoder attends over all the encoded passages to generate the final answer. To make fair comparison of different retrievers, we use the same reader model, FiD-large, to evaluate the retrieved documents from FiD, GAR, and SEAL.

Methods	Index Size	Natural Questions			TriviaQA		
		Top-5	Top-20	Top-100	Top-5	Top-20	Top-100
Dense Retrieval							
DPR	61GB	68.3	80.1	86.1	72.7	80.2	84.8
Lexical Retrieval							
BM25	2.4GB	43.8	62.9	78.3	67.7	77.3	83.9
GAR	2.4GB	60.8	73.9	84.7	71.8	79.5	85.3
SEAL	8.8GB	61.3	76.2	86.3	-	-	-
Ours-single	2.4GB	63.0	75.2	84.8	71.7	79.1	84.6
Ours-multi	2.4GB	**63.9**	**76.8**	**86.7**	**72.3**	**80.1**	**85.8**
Fusion Retrieval							
BM25+DPR	63.4GB	69.7	81.2	88.2	71.5	79.7	85.0
GAR+DPR	63.4GB	72.3	**83.1**	88.9	75.7	82.2	86.3
Ours-single + DPR	63.4GB	72.7	82.6	88.1	76.0	**82.6**	86.4
Ours-multi + DPR	63.4GB	**72.7**	83.0	**89.1**	**76.1**	82.5	**86.4**

Table 4.3: Top-5/20/100 retrieval accuracy (%) and index size (GB) of different models on Natural Questions and TriviaQA test sets. Each score in the right column represents the percentage of the top 20/100 retrieved passages that contain the answers. The DPR and BM25 indexes are downloaded from the Pyserini toolkit[1].

4.4 Results

4.4.1 Contextual Clues Evaluation

We are interested in understanding the quality of the generated contextual clues. In Table 4.2, *Top-1* refers to the top-ranked sequence with the highest probability during beam search, while *Full* contains all top-100 outputs. *Filtered* consists of the final contextual clues after being processed by the filtering step. We report the ROUGE F-measure scores between the ground-truth and generated contextual clues on the NQ validation set. We also report the answer coverage rate, measured as the percentage of contextual clues that contain the answer.

As shown in Table 4.2, rigorously increasing the number of generated candidates increases the ROUGE scores by at least 10% compared with only generating the top sequence, indicating it's more probable to capture the potential ground-truth answer context. The filtering strategy effectively reduces the size of candidate contexts while maintaining high coverage and diversity (less than 1% difference in ROUGE scores). Moreover, *Full* significantly increases the answer coverage rate by $\sim 17\%$ compared with *Top-1*, suggesting that not only more semantics but also more fact words are captured in a larger sizes of candidates.

[1] https://github.com/castorini/pyserini/blob/master/docs/prebuilt-indexes.md

Methods	Index Size	Natural Questions		TriviaQA	
		Top-20	Top-100	Top-20	Top-100
DPR	61GB	80.1	86.1	80.2	84.8
DPR + PCA-256	21GB	77.2	85.5	76.5	83.4
DPR + PCA-256 + PQ	1.3GB	74.8	84.1	74.5	82.6
BPR	2.0GB	77.9	85.7	77.9	84.5
DrBoost	13.5GB	80.9	87.6	-	-
Ours-single	2.4GB	75.2	84.8	79.1	84.6
Ours-multi	2.4GB	76.8	86.7	80.1	85.8

Table 4.4: Comparison with other memory efficient neural retrieval models on index size.

4.4.2 Main Retrieval Results

In Table 4.3, we show both the retrieval accuracy and index size. Note that the index size should be considered with a pinch of salt since it largely depends on the system implementation. The baseline models are reported in their open-sourced versions. Compared with other lexical retrieval models, our method significantly outperforms both GAR and SEAL, showing the effectiveness of extensively sampled contextual clues. We also find that *Ours-multi* consistently improves over *Ours-single*. We surmise that ground-truth answers serve as useful signals during retrieval and they are more likely to be covered when directly sampling answers. Most of the traditional lexical retrieval methods always trail behind dense retrieval by a large margin, as illustrated in Table 4.3. Surprisingly, our method even outperforms the DPR model by 0.6 and 1.0 points in terms of top-100 accuracy on two datasets, while requiring 96% less index storage space. For the purpose of pushing the limit of retrieval performance, we also show the accuracy of different lexical-based methods fused with DPR. Overall, our method fused with DPR achieves the highest accuracy across all baseline methods on both datasets.

We additionally compare our approach with other memory efficient neural retrieval models in Table 4.4. Ma et al. (2021a) show that the DPR could be furthered compressed to trade accuracy off against speed and storage. However, the accuracy of DPR could drop significantly if compressed to the same storage level of the lexical index. BPR (Yamada et al., 2021) integrates a learning-to-hash technique into DPR to represent the passage index using compact binary codes. The index size of BPR is slightly smaller than ours approach, but we achieve higher retrieval accuracy on

Methods	Latency	Natural Questions	
		Top-20	Top-100
DPR	7570ms	**80.1**	**86.1**
DPR + PCA-256	2540ms	77.2	85.5
DPR + PCA-256 + PQ	765ms	74.8	84.1
BM25	318ms	62.9	78.3
GAR (ours)	962ms	73.9	84.7
Ours-single	1545ms	75.2	84.8
Ours-multi	2732ms	**76.8**	**86.7**

Table 4.5: Comparison on retrieval time latency, which is tested using 1 Intel Xeon CPU E5-2699 v4 @ 2.20GHz.

both two datasets. We also include DrBoost (Lewis et al., 2021a), a dense retrieval ensemble trained in stages. DrBoost outperforms ours approach on NQ dataset, while taking 6× times larger index size. Furthermore, we list the latency time in Table 4.5. It is notable that the latency listed in the table is tested on CPU, since we use BM25 as our retrieval backend and it only requires CPU to run. Dense retrieval methods (e.g. DPR) normally is running on GPU devices, which only takes 456.9ms (Lewis et al., 2021a) per query without other device specific speed-up techniques.

4.4.3 End-to-end QA results

As shown in Table 4.6, Ours-multi achieves the highest exact-match scores compared with other baseline methods on both datasets. We have an interesting observation on TriviaQA dataset. The only difference between FiD and Ours is that FiD uses the dense retrieval model DPR, while Ours retrieves from BM25. Considering the Top-100 retrieval accuracy in table 4.3, Ours-single is 0.2 points lower than that of DPR. However, Ours-single increases the EM score on reader by ~ 2 points compared to FiD. It shows that given similar retrieval accuracy, our approach could retrieve qualitatively better passages that are easier for the reader model to answer.

4.5 Conclusion

We propose to narrow the lexical gap between the query and the documents by augmenting the query with extensively sampled contextual clues. To make sure the generated contextual clues are both diverse and relevant, we propose the strategy of

Methods	Natural Questions	TriviaQA
DPR	41.5	57.9
RAG	44.5	56.1
FiD	51.4	67.6
GAR	50.6	70.0
SEAL	50.7	-
Ours-single	50.6	69.7
Ours-multi	**51.7**	**70.8**

Table 4.6: End-to-end exact-match results on the test sets.

context filtering and retrieval fusion. Our approach outperforms both the previous generation-based query expansion method and the dense retrieval counterpart with a much smaller index requirement.

Chapter 5

Apply Flat Minima Optimizers on the Reader Model

The success of modern machine learning in achieving higher performance across diverse tasks can be attributed, in a significant manner, to the increasing complexity of models through overparameterization. This complexity is coupled with the continuous advancement of more potent training algorithms that excel in identifying parameters leading to robust generalization. Ensuring that these models not only perform well on training data but also exhibit generalization beyond this set is of paramount importance. Previous research (Dziugaite and Roy, 2017a; Foret et al., 2021; Izmailov et al., 2018; Jiang et al., 2020; Keskar et al., 2017) extensively explores the interplay between the flatness of minima within the landscape of loss functions and the ability of models to generalize effectively. In light of the potential to enhance generalization by focusing on flatter minima, this chapter aims to concentrate on refining the reader model within the Open-Domain Question Answering (ODQA) pipeline. This refinement is achieved by training the model using optimizers that facilitate the discovery of flatter minima.

Stochastic gradient descent (SGD) methods are central to neural network optimization (Bottou et al., 2018). Recently, one class of algorithms has focused on biasing SGD methods towards so-called '*flat*' minima, which are located in large weight space regions with very similar low loss values (Hochreiter and Schmidhuber, 1997). Theoretical and empirical studies (Bisla et al., 2022; Chaudhari et al., 2017;

Chen et al., 2021b; Dziugaite and Roy, 2017a; Jiang et al., 2020; Keskar et al., 2017; Petzka et al., 2021) postulate that such flatter regions generalize better than sharper minima, e.g., due to the flat minimizer's robustness against loss function shifts between train and test data, as illustrated in Fig 5.1. Two popular flat-minima optimization approaches are: 1. Stochastic Weight Averaging (SWA) (Izmailov et al., 2018), and 2. Sharpness-Aware Minimization (SAM; Foret et al., 2021).

While both strategies aim to find flatter minima, they operate much differently. On the one hand, SWA is based on the intuition that, near a flat minimum, gradients are smaller, leaving many iterates in that flat region. Therefore, averaging iterates will produce a solution that is pulled towards these flatter regions, see 5.1, top. On the other hand, SAM minimizes the maximum loss around a neighborhood of the current iterate. This way, a region around the iterate is designed to have uniformly low loss; see 5.1, bottom. Crucially, SAM requires an additional forward/backward pass for each parameter update, making it more expensive than SWA.

Despite the successes of SWA and SAM in some domains (Athiwaratkun et al., 2019; Bahri et al., 2021; Chen et al., 2021b; Kaddour, 2022; Nikishin et al., 2018), we are unaware of a systematic comparison between them that would help practitioners to choose the right optimizer for their problem and researchers to develop better optimizers. The SWA (Izmailov et al., 2018) paper was published in 2018, and the SAM (Foret et al., 2021) paper in 2021; however, the SAM paper, and its most noticeable follow-ups (Chen et al., 2021b; Kwon et al., 2021; Zhuang et al., 2022), do not compare against SWA. Further, there is very limited overlap in terms of the model architecture and dataset used in the experiments among both papers, which are likely further confounded by other differences in the training procedures (e.g. data augmentations, hyper-parameters, etc.).

The contributions of this chapter are summarized in the following:

1. **In-depth comparison of minima found by SWA and SAM:** We visualize linear interpolations between different models and quantify the minimizers' flatnesses. This analysis yields 4 insights, e.g., despite SAM finding flatter solutions than SWA as quantified by Hessian eigenvalues, they can be close to

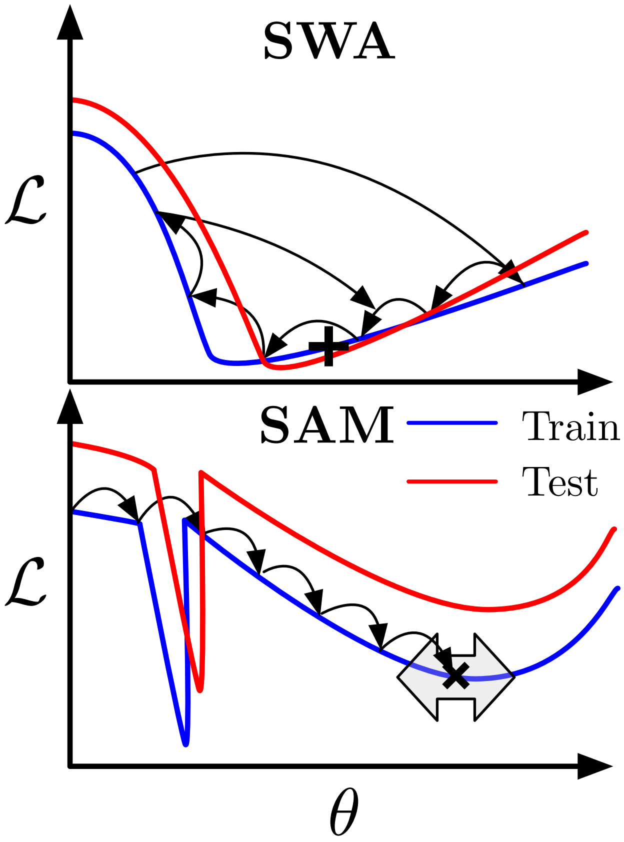


Figure 5.1: The mechanics behind SWA and SAM, whose solution is denoted by+ and ×, respectively. SWA produces a solution θ that is pulled towards flatter regions, while SAM approximates sharpness within the parameters' neighborhood (arrows).

sharp directions, a phenomenon that has been overlooked in the previous SAM literature. Averaging SAM iterates leads to the flattest among all minima.

2. **Rigorous comparison of SWA and SAM's performance over 42 tasks:** We empirically compare the optimizers with a rigorous model selection procedure on a broad range of tasks across different domains (CV, NLP, and GRL), model types (MLPs, CNNs, Transformers) and tasks (classification, self-supervised learning, open-domain question answering, natural language understanding, and node/graph/link property prediction). We discuss 8 findings, e.g., that both dataset and architecture impact their effectiveness, that for NLP tasks, SAM improves over SWA in most cases, and that the converse holds for GRL tasks. When flat-minima optimizers do not help, we notice clear discrepancies between the shapes of loss and accuracy curves. To assist future work, we

open-source the code for all pipelines and hyper-parameters to reproduce the results.

The material in this chapter first appeared in:

*Jean Kaddour, ***Linqing Liu**, Ricardo Silva, and Matt J. Kusner. "When do flat minima optimizers work?." Advances in *Neural Information Processing Systems 35 (2022): 16577-16595.* (*Equal Contribution)

Individual contributions: The genesis of the analysis pertaining to flat minima optimizers stems from the initiative of the first author of this paper. In the experiments, the thesis author conduct all the NLP experiments and subsequent analysis, including part of the CV experiments. The process of paper writing was a collective endeavor undertaken by all contributing authors.

5.1 Background and Related Work

5.1.1 Stochastic Gradient Descent (SGD)

The classic optimization framework of machine learning is empirical risk minimization

$$\mathscr{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \ell(\boldsymbol{x}_i; \boldsymbol{\theta}) \tag{5.1}$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$ is a vector of parameters, $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ is a training set of inputs $\boldsymbol{x}_n \in \mathbb{R}^D$, and $\ell(\boldsymbol{x}; \boldsymbol{\theta})$ is a loss function quantifying the performance of parameters $\boldsymbol{\theta}$ on $\boldsymbol{x}$. SGD samples a minibatch $\mathscr{S} \subset \{1, \ldots, N\}$ of size $|\mathscr{S}| \ll N$ from the training set and updates the parameters through

$$\boldsymbol{\theta}_{t+1}^{\mathrm{SGD}} = \boldsymbol{\theta}_t - \eta \boldsymbol{g}(\boldsymbol{\theta}_t), \text{ where } \boldsymbol{g}(\boldsymbol{\theta}) = \frac{1}{|\mathscr{B}|}\sum_{i \in \mathscr{B}} \nabla \ell(\boldsymbol{\theta}; \boldsymbol{x}_i), \tag{5.2}$$

for a length specified by η, the learning rate.

5.1.2 Stochastic Weight Averaging (SWA)

The idea of averaging weights dates back to accelerating the convergence speed of SGD (Kaddour, 2022; Polyak and Juditsky, 1992). SWA's motivation is based on the following observation about SGD's behavior when training neural networks: it often traverses regions of the weight space that correspond to high-performing models, but rarely reaches the central points of this optimal set. Averaging the parameter values over iterations moves the solution closer to the centroid of this space of points.

The SWA update rule is the cumulative moving average

$$\boldsymbol{\theta}_{t+1}^{\mathrm{SWA}} \leftarrow \frac{\boldsymbol{\theta}_t^{\mathrm{SWA}} \cdot l + \boldsymbol{\theta}_t^{\mathrm{SGD}}}{l+1}, \tag{5.3}$$

where l is the number of distinct parameters averaged so far and t is the SGD iteration number.[1]

SWA has two hyper-parameters: the update frequency ν and starting epoch E.

[1] SWA parameters are constant between averaging steps.

Algorithm 1 Stochastic Weight Averaging ((Izmailov et al., 2018))

```
Input: Loss function $\mathscr{L}$, training budget in number of iterations $b$, training dataset $\mathscr{D} := \cup_{i=1}^{n}\{\boldsymbol{x}_i\}$, mini-batch size $|\mathscr{B}|$, averaging start epoch $E$, averaging frequency $\nu$, (scheduled) learning rate $\eta$, initial weights $\boldsymbol{\theta}_0$.
for $k \leftarrow 1, \dots, b$ do
  Sample a mini-batch $\mathscr{B}$ from $\mathscr{D}$
  Compute gradient $\boldsymbol{g} \leftarrow \nabla\mathscr{L}(\boldsymbol{\theta}_t)$
  Update parameters $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta\boldsymbol{g}$
  if $k \geq E$ and $\mod(k, \nu) = 0$ then
    $\boldsymbol{\theta}_{t+1}^{\text{SWA}} = \left(\boldsymbol{\theta}_t^{\text{SWA}} \cdot l + \boldsymbol{\theta}_{t+1}\right) / (l+1)$
  end if
end for
return $\boldsymbol{\theta}^{\text{SWA}}$
```

Algorithm 2 Sharpness-Aware Minimization ((Foret et al., 2021))

```
Input: Loss function $\mathscr{L}$, training budget in number of iterations $b$, training dataset $\mathscr{D} := \cup_{i=1}^{n}\{\boldsymbol{x}_i\}$, mini-batch size $|\mathscr{B}|$, neighborhood radius $\rho$, (scheduled) learning rate $\eta$, initial weights $\boldsymbol{\theta}_0$.
for $k \leftarrow 1, \dots, b$ do
  Sample a mini-batch $\mathscr{B}$ from $\mathscr{D}$
  Compute worst-case perturbation $\widehat{\boldsymbol{\varepsilon}} \leftarrow \rho\frac{\nabla\mathscr{L}(\boldsymbol{\theta})}{\|\nabla\mathscr{L}(\boldsymbol{\theta})\|_2}$
  Compute gradient $\boldsymbol{g} \leftarrow \nabla\mathscr{L}\left(\boldsymbol{\theta}_t^{\text{SAM}} + \widehat{\boldsymbol{\varepsilon}}\right)$
  Update parameters $\boldsymbol{\theta}_{t+1}^{\text{SAM}} \leftarrow \boldsymbol{\theta}_t^{\text{SAM}} - \eta\boldsymbol{g}$
end for
return $\boldsymbol{\theta}^{\text{SAM}}$
```

When using a constant learning rate, (Izmailov et al., 2018) suggests updating the parameters once after each epoch, i.e. $\nu \approx \frac{N}{|\mathscr{B}|}$, and starting at $E \approx 0.75T$, where T is the training budget required to train the model until convergence with conventional SGD training.

He et al., 2019 argue that SWA may always improve generalization, regardless of the loss function's geometry. Kaddour, 2022 show that averaging a specific range of weights can speed up training convergence. Cha et al., 2021 argue that tuning ν and E carefully is necessary to make it work effectively in domain generalization (DG) tasks. Besides DG tasks, a list of tuned hyper-parameters based on a fair model selection procedure across different architectures and tasks has been missing in the literature. To the best of our knowledge, Cha et al., 2021 is the only study that compares SWA and SAM over the same experiments, but it focuses on domain generalization tasks which we, therefore, leave out in this work.

5.1.3 Sharpness-Aware Minimization (SAM)

While SWA is implicitly biased towards flat minima, SAM *explicitly* approximates the flatness around parameters $\boldsymbol{\theta}$ to guide the parameter update. It first computes the worst-case perturbation $\boldsymbol{\varepsilon}$ that maximizes the loss within a given neighborhood ρ, and then minimizes the loss w.r.t. the perturbed weights $\boldsymbol{\theta}+\boldsymbol{\varepsilon}$. Formally, SAM finds $\boldsymbol{\theta}$ by solving the minimax problem:

$$\min_{\boldsymbol{\theta}} \max_{||\boldsymbol{\varepsilon}||_2 \leq \rho} \mathscr{L}(\boldsymbol{\theta}+\boldsymbol{\varepsilon}), \tag{5.4}$$

where $\rho \geq 0$ is a hyperparameter.

To find the worst-case perturbation $\boldsymbol{\varepsilon}^*$ efficiently in practice, Foret et al., 2021 approximates Eq.5.4 via a first-order Taylor expansion of $\mathscr{L}(\boldsymbol{\theta}+\boldsymbol{\varepsilon})$ w.r.t. $\boldsymbol{\varepsilon}$ around $\mathbf{0}$, obtaining

$$\boldsymbol{\varepsilon}^* \approx \underset{||\boldsymbol{\varepsilon}||_2 \leq \rho}{\arg\max}\, \boldsymbol{\varepsilon}^\top \nabla_{\boldsymbol{\theta}} \mathscr{L}(\boldsymbol{\theta}) \approx \underbrace{\rho \cdot \frac{\nabla_{\boldsymbol{\theta}} \mathscr{L}(\boldsymbol{\theta})}{\|\nabla_{\boldsymbol{\theta}} \mathscr{L}(\boldsymbol{\theta})\|}}_{=:\widehat{\boldsymbol{\varepsilon}}}. \tag{5.5}$$

In words, $\widehat{\boldsymbol{\varepsilon}}$ is simply the scaled gradient of the loss function w.r.t to the current parameters $\boldsymbol{\theta}$. Given $\widehat{\boldsymbol{\varepsilon}}$, the altered gradient used to update the current $\boldsymbol{\theta}_t$ (in place of $\boldsymbol{g}(\boldsymbol{\theta}_t)$) is

$$\nabla_{\boldsymbol{\theta}} \max_{||\boldsymbol{\varepsilon}||_2 \leq \rho} \mathscr{L}(\boldsymbol{\theta}+\boldsymbol{\varepsilon}) \approx \nabla_{\boldsymbol{\theta}} \mathscr{L}(\boldsymbol{\theta})|_{\boldsymbol{\theta}+\hat{\boldsymbol{\varepsilon}}}.$$

Due to Eq.5.5, SAM's computational overhead consists of an additional forward and backward pass per parameter update step compared to SWA and non-flat optimizers.

SAM's performance strongly depends on the neighborhood radius ρ. For example, (Chen et al., 2021b; Wu et al., 2022) show that ρ should be set to values outside the originally considered ranges by (Foret et al., 2021). Analogously to

Sec.5.1.2, this lack of coherence among hyper-parameter tuning protocols in the SAM literature makes it tricky to determine SAM's comparative effectiveness.

5.1.4 Other Flat-Minima Optimizers

There are several extensions of SWA (Cha et al., 2021; Guo et al., 2022) and SAM (Kwon et al., 2021; Zhao et al., 2022; Zhuang et al., 2022). For simplicity, we do not consider them in this work. Besides SWA and SAM, other flat-minima optimizers include e.g. Chaudhari et al., 2017; Dziugaite and Roy, 2017b; Sankar et al., 2020. However, due to their computational cost and/or lack of performance gains, we do not include them in this work. Dziugaite and Roy, 2017b add a regularising term that encourages flat minima based on the PAC-Bayes bound for generalisation (Pérez-Ortiz et al., 2021). Chaudhari et al., 2017 requires $[5,20]$ forward and backward passes per parameter update. Sankar et al., 2020 similarly requires $[5,10]$ forward and backward passes to estimate the Hessian trace and 6 of 7 experiments yield minimal improvement of $\leq 0.27\%$, see Table 1 in (Sankar et al., 2020). In contrast, SWA and SAM have been shown to increase performance by multiple percentage points in some cases (Cha et al., 2021; Chen et al., 2021b), while requiring many fewer computational resources.

5.2 How do minima found by SWA and SAM differ?

In this section, we investigate SWA and SAM solutions for two prototypical deep learning tasks, where these optimizers improve over the baseline. Our goal is to better understand their geometric properties.

First, we investigate the behavior of the loss landscape along the line between non-flat and flat solutions (Sec.5.2.1). Previous studies successfully used such linear interpolations to gain novel insights, e.g., for training dynamics (Frankle, 2020; Goodfellow and Vinyals, 2015), regularization (Geiping et al., 2021; Li et al., 2018), and network pruning (Frankle et al., 2020). Second, motivated by findings in Sec.5.2.1, we average SAM iterates and visualize interpolations between averaged and non-averaged solutions (Sec.5.2.2). Interestingly, the averaged SAM solution is less susceptible to asymmetric directions. Third, we compare quantitative measurements of all solutions' flatnesses (Sec.5.2.3). Here, we compute dominant Hessian eigenvalues, as commonly used in the flat minima literature (Chaudhari et al., 2017; Chen et al., 2021b; Foret et al., 2021; Yao et al., 2020).

We choose the following two disparate learning settings: (i) a well-known image classification task, widely used for evaluation in flat-minima optimizer papers, and (ii) a novel, challenging Python code summarization task, on which state-of-the-art models achieve only around 16% F1 score on the test set (which is **higher** than its commonly achieved accuracy on the more challenging training set), and that has not been explored yet in the flat-minima literature. Specifically, for (i), we investigate the loss/accuracy surfaces of a WideResNet28-10 (Zagoruyko and Komodakis, 2016) model on CIFAR-100 (Krizhevsky, 2009) (baseline non-flat optimizer: SGD with momentum (SGD-M)) (Rumelhart et al., 1988). For (ii), we use the theoretically-grounded Graph Isomorphism Network (Xu et al., 2019) model on OGB-Code2 (Hu et al., 2020) (baseline optimizer: Adam (Kingma and Ba, 2015a)).

All optimizers start from the same initialization. We denote the minimizer produced by the non-flat methods (SGD-M and Adam) by $\boldsymbol{\theta}^{\mathrm{NF}}$ and the flat ones by $\boldsymbol{\theta}^{\mathrm{SWA}}$ and $\boldsymbol{\theta}^{\mathrm{SAM}}$.

5.2.1 What is between non-flat and flat solutions?

We start by comparing the similarity of flat and non-flat minimizers through linear interpolations. This analysis allows us to understand if they are in the same basin, and how close they are to a region of sharply-increasing loss, where we expect loss/accuracy to differ widely between train and test. Further, for each of our 4 observations, we recommend a future work direction.

To linearly interpolate between two sets of parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, we parameterize the line connecting these two by choosing a scalar parameter α, and defining the weighted average $\boldsymbol{\theta}(\alpha) = (1-\alpha)\boldsymbol{\theta} + \alpha\boldsymbol{\theta}'$. If there exists no high-loss barrier between two networks $\boldsymbol{\theta}, \boldsymbol{\theta}'$ along the linear interpolation, we say that they are located in the same *basin*, i.e., $\{\boldsymbol{\theta}, \boldsymbol{\theta}'\} \in \boldsymbol{\Omega}$. (Neyshabur et al., 2020; Zhou et al., 2020). A basin is an area in the parameter space where the loss function has relatively low values. Due to NN non-linearities, the linear combination of the weights of two accurate models does not necessarily define an accurate model. Hence, we generally expect high-loss barriers along the linear interpolation path.

While there are alternative distance measures that could be used to compare two networks, they typically either (a) do not offer clear interpretations, as pointed out by (Frankle et al., 2020), or (b) yield trivial network connectivity results, such as *non-linear* low-loss paths, which can be found for any two network minimizers (Draxler et al., 2018; Fort and Jastrzebski, 2019; Garipov et al., 2018; Gotmare et al., 2019).

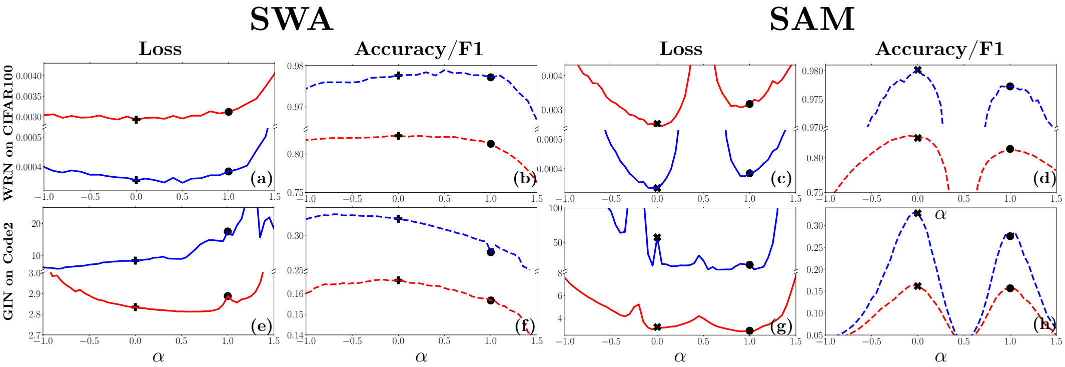


Figure 5.2: Training (blue) and test (red) losses (—) and accuracies of linear interpolations $\boldsymbol{\theta}(\alpha) = (1-\alpha)\boldsymbol{\theta} + \alpha\boldsymbol{\theta}'$ (for $\alpha \in [-1, 1.5]$) between SWA (+) and SAM (×) solutions ($\alpha = 0.0$) and non-flat baseline solutions ($\bullet, \alpha = 1.0$).

Obs. 1: $\{\boldsymbol{\theta}^{\mathbf{SWA}}, \boldsymbol{\theta}^{\mathbf{NF}}\} \in \boldsymbol{\Omega}^{\mathbf{NF}}$. $\boldsymbol{\theta}^{\text{SWA}}$ and $\boldsymbol{\theta}^{\text{NF}}$ are in the same basin, as can be seen in Figures 5.2a and 5.2e. Additionally, $\boldsymbol{\theta}^{\text{NF}}$ is near the periphery of a sharp increase in loss, as can be seen when moving in the direction from $\boldsymbol{\theta}^{\text{SWA}}$ to $\boldsymbol{\theta}^{\text{NF}}$ (i.e., $\alpha > 1$). Conversely, $\boldsymbol{\theta}^{\text{SWA}}$ finds flat regions that change slowly in the loss. This bias of SWA to flatter loss beneficially transfers to the accuracy landscape too: Figures 5.2b and 5.2f show the accuracy/F1 score rapidly dropping off approaching and beyond $\boldsymbol{\theta}^{\text{NF}}$. Interestingly, in Figures 5.2e and 5.2f, we see that for Code2, for $\alpha < 0$, there exist solutions with even better training loss/accuracy but worse test loss/accuracy. However, $\boldsymbol{\theta}^{\text{SWA}}_{\text{GIN}}$ is close to the test accuracy maximizer along this interpolation. Future work may inspect why the cross entropy loss function used for GIN/Code2 seems less well correlated with its accuracy compared to WRN/CIFAR100.

Obs. 2: $\boldsymbol{\theta}^{\mathbf{SAM}} \in \boldsymbol{\Omega}^{\mathbf{SAM}} \neq \boldsymbol{\Omega}^{\mathbf{NF}}$. $\boldsymbol{\theta}^{\text{SAM}}$ and $\boldsymbol{\theta}^{\text{NF}}$ are not in the same basin:

Figures 5.2c and 5.2g show that there is a high loss barrier between them, respectively. Figures 5.2d and 5.2h show that $\boldsymbol{\theta}^{\text{SAM}}$ and even nearby points in parameter space achieve higher accuracies/F1 scores (i.e. generalize better) than $\boldsymbol{\theta}^{\text{NF}}$ and points around it. This is an interesting result because we expect different basins to produce qualitatively different predictions, one of the motivations behind combining models, even if they exhibit different performances (Huang et al., 2017; Lakshminarayanan et al., 2017). Grewal and Bui, 2021 successfully combine models yielded by different optimizers, and we think future work should study ensembling SAM and non-SAM solutions.

Obs. 3: SAM finds a saddle point. Figure 5.2g shows $\boldsymbol{\theta}^{\text{SAM}}_{\text{GIN}}$ being located in a sharp training loss minimum whose loss is much higher than $\boldsymbol{\theta}^{\text{NF}}$. Yet, its test loss is only slightly higher, and its F1 score is better. We visualize 2D plots moving along random directions (not shown here due to space) to confirm that $\boldsymbol{\theta}^{\text{SAM}}_{\text{GIN}}$ is a saddle point. A common pathology among curvature-based methods is that they attract saddle points (Dauphin et al., 2014). Since SAM takes some form of curvature into account, too, we believe that future work should investigate SAM's propensity to find saddle points and potential remedies.

Obs. 4: $\boldsymbol{\theta}^{\text{SAM}}$ is closer to sharper directions than $\boldsymbol{\theta}^{\text{SWA}}$, as can be seen by $\mathscr{L}_{\text{tr/te}}(\boldsymbol{\theta}^{\text{SAM}}(0.1)) \approx 2 \cdot \mathscr{L}_{\text{tr/te}}(\boldsymbol{\theta}^{\text{SAM}}(-0.1))$, while $\mathscr{L}_{\text{tr/te}}(\boldsymbol{\theta}^{\text{SWA}}(0.1)) \approx \mathscr{L}_{\text{tr/te}}(\boldsymbol{\theta}^{\text{SWA}}(-0.1))$, where $\mathscr{L}(\cdot)_{\text{tr/te}}$ refers to both training and test loss functions. A possible explanation for SAM being closer to sharp sides is that while it finds different basins than SGD/SWA by smoothing the loss surface (as illustrated in Fig.5.1), *within* a local basin, it may oscillate around the minimizer similarly as SGD. One cause for this can be that $\boldsymbol{\Omega}^{\text{SAM}}$'s hypersphere is larger than SAM's radius ρ. If that holds, then given a small enough learning rate, we expect it to oscillate around $\boldsymbol{\theta}^* \in \boldsymbol{\Omega}^{\text{SAM}}$ (the smaller the learning rate, the less likely it escapes the basin due to that stochasticity). Two possible remedies for that are: (1) adapt/schedule ρ, or (2) average SAM iterates to bias its solution towards the flatter side. (1) has been explored by (Zhao et al., 2022; Zhuang et al., 2022). We try (2) in the next subsection. Future work may study SAM's basin escape time, e.g., using convolutions (Kleinberg et al., 2018) or stochastic differential equations (Zhou et al., 2020).

5.2.2 What happens if we average SAM iterates?

Based on observation 4: "**$\boldsymbol{\theta}^{\text{SAM}}$ is closer to sharper directions than $\boldsymbol{\theta}^{\text{SWA}}$**", averaging SAM iterates may further improve generalization, referred to as *Weight-Averaged Sharpness-Aware Minimization* (WASAM). The reason is that while SAM finds better-performing basins, *within* the basin, its final iterate may still be near a side that increases sharply in the loss.

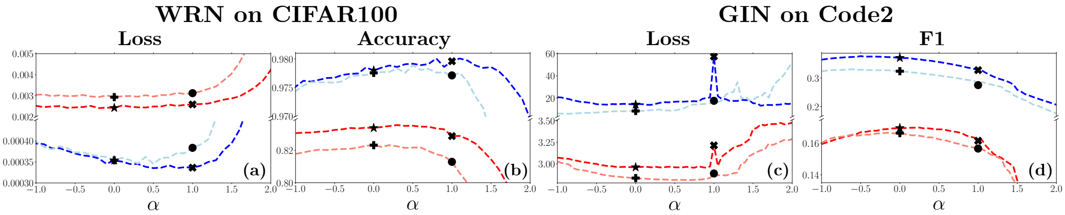


Figure 5.3: Training (blue) / test (red) losses (—) and accuracies between non-flat baseline (●) ↔ SWA (+), SAM (×) ↔ WASAM (⋆). $\alpha = 0.0$ refers to SAM solutions, $\alpha = 1.0$ refers to non-flat baseline solutions.

Starting with the first of the two previously analyzed settings (WRN/CIFAR100), Figures 5.3a, and 5.3b show that $\boldsymbol{\theta}^{\text{WASAM}}_{\text{WRN}}$ (marker: ★) achieves the lowest test loss and highest test accuracy, respectively. What stands out in comparison to the previous plots is $\boldsymbol{\theta}^{\text{SAM}}_{\text{WRN}}$'s (×) proximity to sharp sides, surprisingly similar to $\boldsymbol{\theta}^{\text{NF}}_{\text{WRN}}$ (●) here and in Figures 5.2c and 5.2e. As we hoped, $\boldsymbol{\theta}^{\text{WASAM}}_{\text{WRN}}$ is indeed closer to a flatter region, as can be seen by $\mathscr{L}_{\text{tr/te}}(\boldsymbol{\theta}^{\text{WASAM}}_{\text{WRN}}(-0.2)) \approx \mathscr{L}_{\text{tr/te}}(\boldsymbol{\theta}^{\text{WASAM}}_{\text{WRN}}(0.2))$.

In GIN/OGB-Code2, one unanticipated finding is that $\boldsymbol{\theta}^{\text{WASAM}}_{\text{GIN}}$ escapes the (previously discussed) saddle point of $\boldsymbol{\theta}^{\text{SAM}}_{\text{GIN}}$, appearing here as a maximum in Figure 5.3c. A likely reason for that is that SAM traversed nearby flatter regions before arriving at the saddle point, especially if it is a non-strict saddle. In terms of F1 score, Figure 5.3d shows that while $\boldsymbol{\theta}^{\text{SWA}}_{\text{GIN}}$ (+) and $\boldsymbol{\theta}^{\text{SAM}}_{\text{GIN}}$ perform about equally well, the flatter region found by $\boldsymbol{\theta}^{\text{WASAM}}_{\text{GIN}}$ improves over both.

5.2.3 How "flat" are the found minima?

We now quantify the flatnesses of all four optimizers over both tasks by computing the median of the dominant Hessian eigenvalue across all training set batches using

Task	Baseline	SWA	SAM	WASAM
WRN on CIFAR100	673	265	237	**117**
GIN on Code2	16.65	16.79	11.31	**9.96**

Table 5.1: Median $\boldsymbol{\lambda}_{\max}$ of Hessian over all training set batches.

the Power Iteration algorithm (Mises and Pollaczek-Geiringer, 1929; Yao et al., 2020). This metric measures the worst-case loss landscape curvature. We choose this metric as it is very commonly used in the minima flatness literature, e.g., (Chaudhari et al., 2017; Chen et al., 2021b; Dong et al., 2019; Foret et al., 2021; Krishnapriyan et al., 2021; Stutz et al., 2021; Yao et al., 2019).

Table 5.1 shows that SAM leads to flatter minima than SWA in both cases. Interestingly $\boldsymbol{\lambda}_{\max}(\boldsymbol{\theta}^{\mathrm{NF}}_{\mathrm{WRN}}) \approx 2.5 \cdot \boldsymbol{\lambda}_{\max}(\{\boldsymbol{\theta}^{\mathrm{SWA}}, \boldsymbol{\theta}^{\mathrm{SAM}}\})$, while $\boldsymbol{\lambda}_{\max}(\boldsymbol{\theta}^{\mathrm{NF}}_{\mathrm{WRN}}) \approx 5.75\boldsymbol{\lambda}_{\max}(\boldsymbol{\theta}^{\mathrm{WASAM}}_{\mathrm{WRN}})$, indicating room for improvement in terms of flatness for both SWA and SAM. The relative differences are less dramatic for GIN/Code2, although surprisingly $\boldsymbol{\lambda}_{\max}(\boldsymbol{\theta}^{\mathrm{NF}}_{\mathrm{GIN}}) \approx \boldsymbol{\lambda}_{\max}(\boldsymbol{\theta}^{\mathrm{SWA}}_{\mathrm{GIN}})$. In sum, averaging SAM iterates leads to the flattest minima **and** best-performing minima in both cases (see Sec.5.3).

5.3 How do SWA and SAM perform on the NLP tasks?

As we point out in the introduction, there is almost no overlap and consistency regarding reported SWA and SAM results in the literature. This section addresses this gap. For example, Bahri et al., 2021; Chen et al., 2021b illustrate that the flat minima found by SAM improve generalization on Transformer (Vaswani et al., 2017a) architectures compared to non-flat optimizers, but they do not compare against SWA. Hence, it is unclear if the computationally cheaper SWA may provide better or similar performance.

5.3.1 Experiment Setup

We compare flat minimizers SWA, SAM, and averaged SAM iterates (WASAM) over the non-flat minimizers across a range of different tasks in the domains of computer vision, natural language processing, and graph representation learning. We average all runs at least three times across random seeds. We bold the best-performing approach and any approach whose average performance plus standard error overlaps it. In this thesis, we will focus on the task of Open Domain Question Answering.

For all architectures and datasets, we set hyperparameters shared by all methods (e.g., learning rate) mostly to values cited in prior work [2] As explained in Sec.5.1.2 and Sec.5.1.3, the effectiveness of flat-minima optimizers is highly sensitive to their additional hyper-parameters. We select hyper-parameters using a grid search over a held-out validation set. Specifically, for SWA we follow Izmailov et al., 2018 and hold the update frequency ν constant to once per epoch and tune the start time $E \in \{0.5T, 0.6T, 0.75T, 0.9T\}$ (T is the number of baseline training epochs). Izmailov et al., 2018 argue that a cyclical learning rate starting from E helps to encourage exploration of the basin. For the sake of simplicity, we average the iterates of the baseline directly but include even earlier starting times (i.e., $0.5T, 0.6T$). For SAM, we tune its neighborhood size $\rho \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$, as in previous work (Bahri et al., 2021; Foret et al., 2021).

For the task of Open Domain Question Answering, we adapt the hyper-parameter values and the top-25 retrieved passages for each question from Izacard and Grave, 2021. We report the Exact Match score of FiD-base model on Natural Questions (NQ) (Kwiatkowski et al., 2019a) and TriviaQA (Joshi et al., 2017) test sets.

For GLUE benchmark, we report Matthew's Corr for CoLA, Pearson correlation coefficient for STSB, and accuracy for the the rest of the datasets. Results are all evaluated on the dev set of GLUE benchmark. We use the RoBERTa-base as our backbone language model, implemented with Huggingface Transformers (Wolf et al., 2020). Most of the task-specific hyper-parameter values are adapted from Aghajanyan et al., 2020.

[2] Sometimes with minor modifications, e.g., adjusting per-device batch sizes to be compatible with our GPU infrastructure.

Task	Model	Baseline	SWA	SAM	WASAM
NQ	FiD	$49.35_{\pm 0.44}$	$-0.20_{\pm 0.33}$	$+\,\mathbf{0.33}_{\pm \mathbf{0.19}}$	$+\,\mathbf{0.48}_{\pm \mathbf{0.21}}$
TriviaQA	FiD	$67.74_{\pm 0.29}$	$+\,0.40_{\pm 0.24}$	$+\,\mathbf{0.89}_{\pm \mathbf{0.03}}$	$+\,\mathbf{0.92}_{\pm \mathbf{0.10}}$
COLA	RoBERTa	$60.41_{\pm 0.22}$	$+\,0.09_{\pm 0.08}$	$+\,\mathbf{1.57}_{\pm \mathbf{1.20}}$	$+\,\mathbf{1.41}_{\pm \mathbf{1.14}}$
SST	RoBERTa	$94.95_{\pm 0.13}$	$-0.30_{\pm 0.27}$	$-0.23_{\pm 0.40}$	$+\,\mathbf{0.19}_{\pm \mathbf{0.14}}$
MRPC	RoBERTa	$89.14_{\pm 0.57}$	$+\,0.08_{\pm 0.49}$	$\mathbf{+0.73}_{\pm \mathbf{0.43}}$	$+\,\mathbf{0.81}_{\pm \mathbf{0.38}}$
STSB	RoBERTa	$90.40_{\pm 0.02}$	$+0.00_{\pm 0.05}$	$\mathbf{+0.38}_{\pm 0.17}$	$+\,\mathbf{0.35}_{\pm \mathbf{0.16}}$
QQP	RoBERTa	$91.36_{\pm 0.07}$	$+0.01_{\pm 0.06}$	$\mathbf{+0.08}_{\pm \mathbf{0.07}}$	$\mathbf{+0.06}_{\pm \mathbf{0.08}}$
MNLI	RoBERTa	$87.41_{\pm 0.09}$	$+0.08_{\pm 0.11}$	$\mathbf{+0.39}_{\pm \mathbf{0.02}}$	$+0.35_{\pm 0.03}$
QNLI	RoBERTa	$92.96_{\pm 0.06}$	$-0.08_{\pm 0.11}$	$+0.09_{\pm 0.01}$	$\mathbf{+0.11}_{\pm \mathbf{0.06}}$
RTE	RoBERTa	$80.09_{\pm 0.23}$	$-0.23_{\pm 0.20}$	$\mathbf{+0.70}_{\pm \mathbf{0.65}}$	$-0.46_{\pm 0.12}$

Table 5.2: Experiment results on the test sets of NQ and TriviaQA datasets, dev sets of the GLUE benchmark.

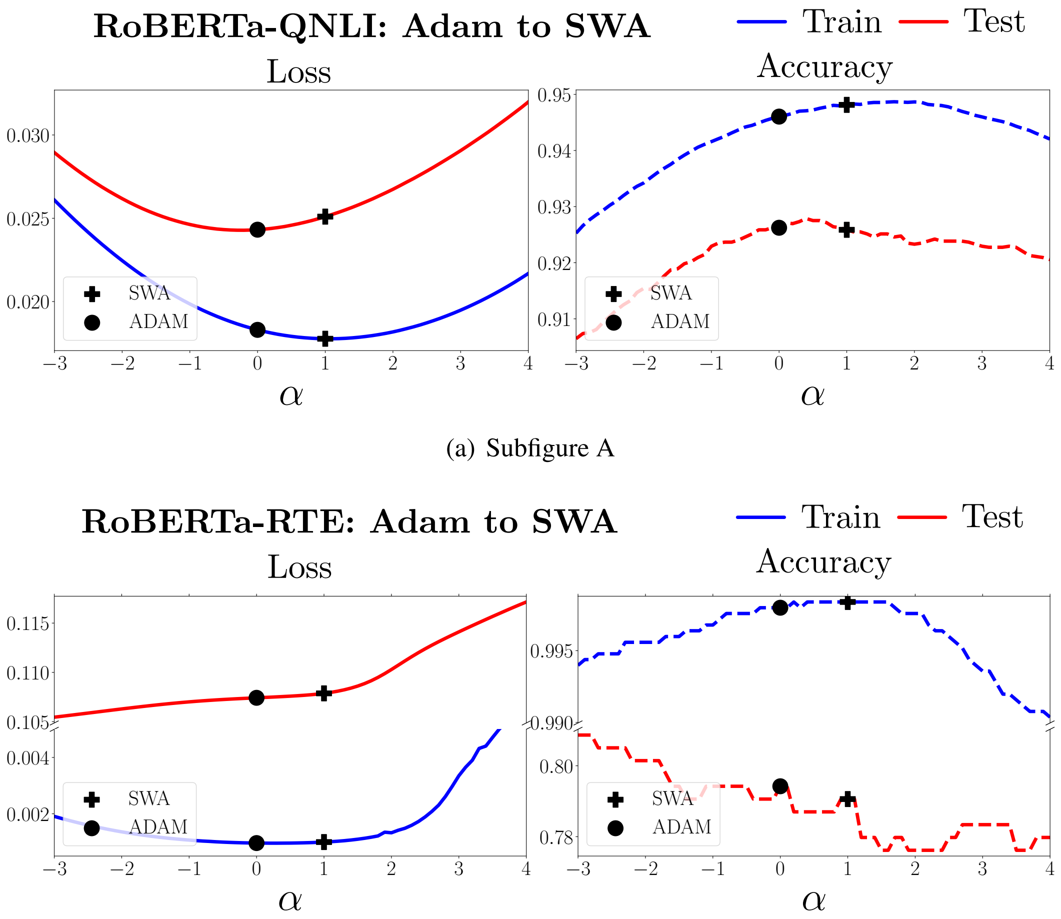
5.3.2 Results

The experiments results are included in Table 5.2. We observe that across different tasks, the baseline non-flat optimizer and SWA are never among the most accurate. SWA often does not improve and sometimes hurts performances. Both SAM and WASAM are the best in all but two (different) cases, only one of which is worse than the baseline. Averaging SAM iterates (WASAM) often improves over SWA or SAM alone. We hypothesize that asymmetric payoffs are the reason: when either SWA or SAM does not improve over the baseline (as discussed above), it does not hurt (much) either, hence WASAM is more robust across all tasks.

5.3.3 Why does SWA not work well on NLP tasks?

In Table 5.2, we saw that SWA had only a mild effect on the generalization performance of NLP tasks, and sometimes even decreases it. Here, we seek to investigate why that is.

We consider two tasks: (i) the RTE task, for which SWA decreases the performance by around $-0.23_{\pm 0.20}$ compared to Adam, (ii) the QNLI task, for which SWA decreases the performance by $-0.08_{\pm 0.11}$. In both cases, SAM improved the performance statistically significantly over the baseline optimizer Adam.



(a) Subfigure A

(b) Subfigure B

Figure 5.4: Training (blue) and test (red) losses (—) and accuracies of linear interpolations $\boldsymbol{\theta}(\alpha) = (1-\alpha)\boldsymbol{\theta} + \alpha\boldsymbol{\theta}'$ between Adam solutions ($\bullet, \alpha = 0.0$) and SWA ($+, \alpha = 1.0$).

For the QNLI task in upper Fig. 5.4, we observe that SWA finds a lower/higher training loss/accuracy than Adam, respectively. However, the test loss/accuracy is higher/lower at the SWA solutions and the loss functions seem less well correlated in between both solutions (i.e, for $\alpha \in [0,1]$).

For the RTE task in lower Fig. 5.4, we note that SWA finds a solution that is closer to a sharply increasing side. This may happen if the baseline optimizer skips or goes around sharper solutions (e.g., due to large step sizes) and the average pulls it towards these suboptimal regions.

Furthurmore, we compute the Centered Kernel Alignment Similarities (CKA; Kornblith et al., 2019; Yang et al., 2021) and cosine similarities of network output logits on train and test set, respectively. Table 5.3 shows the results. CKA is a similarity index that measures the similarities between deep neural network representations.

Table 5.3: Pairwise CKA (Kornblith et al., 2019) and cosine similarities between non-flat (NF) and SWA/SAM solutions. SWA solutions produce predictions more similar to NF ones than SAM.

Task	$s_{\text{CKA}}(\boldsymbol{\theta}^{\text{NF}}, \boldsymbol{\theta}^{\text{SWA}})$	$s_{\text{cosine}}(\boldsymbol{\theta}^{\text{NF}}, \boldsymbol{\theta}^{\text{SWA}})$	$s_{\text{CKA}}(\boldsymbol{\theta}^{\text{NF}}, \boldsymbol{\theta}^{\text{SAM}})$	$s_{\text{cosine}}(\boldsymbol{\theta}^{\text{NF}}, \boldsymbol{\theta}^{\text{SAM}})$
RoBERTa-QNLI (Train)	0.9997	0.9991	0.9790	0.9510
RoBERTa-QNLI (Valid)	0.9830	0.9959	0.9550	0.9530
RoBERTa-RTE (Train)	0.9931	0.9891	0.9831	0.9628
RoBERTa-RTE (Test)	0.9314	0.9567	0.8808	0.8927

The results show that the SAM solutions produce predictions that are less similar to the non-flat baseline than SWA solutions, as indicated by lower CKA and cosine similarities. This result is in line with Observation 1 and 2 from Sec. 5.2.1.

5.4 Limitations and Future Work

First, some of the fixed, shared hyperparameter values we used from previous works may harm the effect of flat optimizers. The ideal experimental design includes tuning all hyperparameters independently for the non-flat optimizer, SWA, SAM, and WASAM. However, this forces the number of required runs to grow exponentially in unique hyperparameters and quickly renders this benchmark infeasible.

Second, despite our best efforts to evaluate the optimizers on a broad range of benchmark tasks, there are still plenty of unexplored domains; especially some of which are known to be sensitive to careful optimization, such as generative modeling (Heusel et al., 2017), deep reinforcement learning (Arulkumaran et al., 2017), or causal machine learning (Kaddour et al., 2022).

Third, in general, we believe fruitful directions of research include (a) optimizers that explicitly find basins where training loss flatness more directly corresponds to higher hold-out accuracy, (b) post-processing methods for existing optimization runs to move into flatter regions of these basins, (c) loss functions whose contours more tightly align with accuracy contours, (d) the study of flat-minima hyperparameter interactions (e.g., learning rate and neighborhood radius in SAM), (e) analyses of flat minima optimization on convergence speed (Kaddour, 2022).

Our benchmark results point to which tasks would most benefit from improving these future work directions: graph learning tasks would clearly benefit from improvements in (a), as SAM is never among the best performing method, and language tasks would benefit if (b) is improved, as SWA is never among the best performing method.

5.5 Conclusion

We investigated when flat minimia optimizers work by conducting a fair comparison of two popular flat-minima optimizers. We examined the behavior of SWA/SAM by analyzing their loss landscapes on two representative deep learning tasks. Our next step was to evaluate their generalization performance on a broad and diverse set of tasks (in data, learning settings, and model architectures). Based on this benchmarking, our findings directly guide future work directions. Finally, when SWA/SAM did not improve over baselines, common assumptions seemed broken (i.e., train-to-test loss minimizers were not correlated).

Chapter 6

Conclusions and Future Work

6.1 Main Contributions

In this thesis, our investigation revolves around enhancing the generalization capability of Open Domain Question Answering (ODQA) systems. These systems are designed to tackle novel questions that lie beyond their training data. In Chapter 1, we lay the foundation by presenting fundamental definitions of the ODQA task. We delve into the essential steps required to address input questions and delineate the various dimensions that encompass generalization within the ODQA context.

Moving forward to Chapter 2, we delve into the contextual backdrop and related research. We embark on a historical journey, tracing the evolution of methodologies in ODQA. We extensively explore three primary approach paradigms: Retriever-Reader, QA-pair retriever, and Parametric models. Additionally, we delve into the metrics employed for evaluating these approaches.

The intricacies of the challenges plaguing the model's suboptimal performance on novel questions take center stage in Chapter 3. Through an innovative dataset annotation methodology, we classify questions based on their degree of generalization. This categorization facilitates the identification and quantification of key factors influencing the model's generalization prowess.

Guided by insights derived from Chapter 4, we propose a novel enhancement for the document retrieval phase by incorporating contextual clue sampling via Language Models. This augmentation not only enhances retrieval accuracy but also substantially reduces the requisite index size and retrieval latency.

Furthermore, our focus shifts to refining the reader module in Chapter 5. Building on prior research that underscores the interplay between the topographical landscape of loss functions and a model's generalization capacity, we endeavor to train our model using optimizers conducive to discovering flatter minima. Empirical evidence is marshaled to corroborate the efficacy of training models with flat minima optimizers in boosting reader model performance.

We specify the major contributions and findings below:

Novel Dataset Annotation Methods: We propose to annotate the existing ODQA dataset in order to understand which aspects of novel questions make them challenging for the model to answer. Drawing upon studies on systematic generalization, we introduce and annotate questions according to three categories that measure different levels and kinds of generalization: training set overlap, compositional generalization (comp-gen), and novel-entity generalization (novel-entity). Our categorization breakdown is motivated by how they capture different levels of generalization: overlap requiring no generalization beyond recognizing paraphrases, comp-gen requiring generalization to novel compositions of previously observed entities and structures, and novel-entity requiring generalization to entities not present in the training set.

Identifying Factors Influencing Model Generalization: We demonstrate and quantify key factors that impact model generalization performance: cascading errors from the retrieval component, frequency of question pattern, and frequency of the entity. Notably, we uncover that the inherent structure of questions itself can pose challenges, enabling us to deduce overarching question patterns through our decomposition approach. We find a strong positive correlation between the pattern frequency in the training set and test accuracy. Our investigation extends to the behavior of non-parametric models in handling distinct question subsets comp-gen and novel-entity cases. Surprisingly, for comp-gen questions, we identify a strong negative correlation between the frequency of mentioned entities within a question and test accuracy. For novel-entity questions, substituting novel entities with those from the training set maintains performance, suggesting that specific unseen entities aren't the primary performance bottleneck. Instead, it suggests a potential limitation in the model's compositional generalization. Moving beyond questions, our analysis delves into retrieved passages. We observe a shared deficiency in retrieval accuracy for both the comp-gen and novel-entity subsets, hovering around 75% for top-20 accuracy. Intriguingly, many passages containing correct answers lack sufficiently informative context for the reader model to precisely locate them. This underscores the need to enhance the reader's capacity to reason across multiple passages or for the retriever model to furnish passages with richer contextual information.

Query Expansion Using Language Models: Query expansion is an effective approach for mitigating vocabulary mismatch between queries and documents in information retrieval. One recent line of research uses language models to generate query-related contexts for expansion. Along this line, we argue that expansion terms from these contexts should balance two key aspects: diversity and relevance. The obvious way to increase diversity is to sample multiple contexts from the language model. However, this comes at the cost of relevance, because there is a well-known tendency of models to hallucinate incorrect or irrelevant contexts. To balance these two considerations, we propose a combination of an effective filtering strategy and fusion of the retrieved documents based on the generation probability of each context. Our lexical matching based approach achieves a similar top5/top-20 retrieval accuracy and higher top-100 accuracy compared with the well-established dense retrieval model DPR, while reducing the index size by more than 96%. For end-to-end QA, the reader model also benefits from our method and achieves the highest Exact-Match score against several competitive baselines.

Understanding Flat Minima Optimizers: Recently, flat-minima optimizers, which seek to find parameters in low-loss neighborhoods, have been shown to improve a neural network's generalization performance over stochastic and adaptive gradient-based optimizers. Two methods have received significant attention due to their scalability: 1. Stochastic Weight Averaging (SWA), and 2. Sharpness-Aware Minimization (SAM). However, there has been limited investigation into their properties and no systematic benchmarking of them across different domains. We fill this gap here by comparing the loss surfaces of the models trained with each method and through broad benchmarking across computer vision, natural language processing, and graph representation learning tasks. We discover several surprising findings from these results, which we hope will help researchers further improve deep learning optimizers, and practitioners identify the right optimizer for their problem.

Training Reader model with Flat Minima Optimizers: The existing literature presents a notable scarcity of both overlap and consistency in the reported outcomes of SWA and SAM, the two most widely used flat minima optimizers. In response, we undertake a comprehensive array of experiments spanning diverse NLP tasks, encompassing ODQA and the GLUE benchmark. Ensuring the robustness of our findings, we meticulously average the results from multiple runs conducted with varying random seeds. Drawing from the outcomes of these experiments, intriguing patterns emerge. Across the spectrum of tasks, neither the conventional non-flat optimizer nor SWA manages to consistently claim the mantle of highest accuracy. SWA, in particular, exhibits a propensity to underdeliver in terms of performance improvements, and at times, it even leads to performance deterioration. In stark contrast, both SAM and WASAM consistently shine, outperforming their counterparts in all but two distinct cases. Impressively, the practice of iteratively averaging SAM, exemplified by WASAM, frequently surpasses the performance of standalone SWA or SAM. A noteworthy highlight pertains to the reader module FiD model (Izacard and Grave, 2021), exclusively trained using SAM. This model achieves competitive performance compared to its counterpart trained with baseline optimizers, all while requiring just 20 retrieved documents as input, as opposed to the 100 documents needed by the baseline model. Furthermore, our inquiry extends to unraveling the reasons behind SWA's suboptimal performance in NLP tasks.

6.2 Limitations and Future Work

6.2.1 Exploring recent released large language models

The experiments conducted in this thesis is built on the relatively early and small-scale language models, such as BART-large (400M paramters) and T5-large (770M parameters). However, a very recent surge of remarkable endeavors has led to the creation of colossal models, boasting tens to hundreds of billions of parameters. These monumental models have taken the spotlight in the realm of newly established products, including but not limited to ChatGPT by OpenAI[1], Claude by Anthropic[2], Github Copilot[3], and Character chatbots[4]. These models have demonstrated unprecedented levels of performance across a myriad of downstream tasks, marking a significant advancement. As we delve into the context of Open-Domain Question Answering (ODQA), it becomes imperative to thoroughly evaluate and comprehend their performance within this domain.

Many efforts have already been made to measure the large language models. Liang et al. (2022) propose the Holistic Evaluation of Language Models (HELM) across 30 prominent language models on all 42 scenarios, including 21 scenarios that were not previously used in mainstream LM evaluation. In the realm of 9 core question answering situations, InstructGPT davinci v2 (175B) stands as the most precise model across all 9 scenarios. Intriguingly, among the top 3 models in terms of accuracy for 6 out of the 9 scenarios, there isn't a publicly available model. The ranking of accuracy generally features InstructGPT davinci v2 (175B) at the forefront, succeeded by Anthropic-LM v4-s3 (52B) and TNLG v2 (530B) in a descending hierarchy. Liu et al. (2023a) analyze language model performance on tasks that require identifying relevant information within their input contexts (multi-document question answering and key-value retrieval). Their findings reveal that optimal performance frequently emerges when pertinent information is located at the outset or conclusion of the input context. Conversely, the effectiveness notably diminishes when models are required to retrieve relevant details from the middle of extensive contexts. Moreover, as the input context extends in length, there is a significant decline in performance, even among models explicitly designed for handling lengthy contexts. In contrast to the assessment of a model's capacity to effectively utilize context-based information, an alternative avenue of research proposes the integration of language

[1] https://openai.com/chatgpt

[2] https://www.anthropic.com/index/introducing-claude

[3] https://github.com/features/copilot

[4] https://beta.character.ai/

models that furnish references to substantiate their generated content. Generative search engines embody this approach by producing responses to input queries while also providing inline citations. Notably, Bing Chat from Microsoft served 45 million chats in the first month of its public review (Mehdi, 2023). Additional commercial generative search engines, such as Perplexity.ai[5], YouChat[6], and NeevaAI[7] are also in the landscape. In a recent study by Liu et al. (2023b), a comprehensive human evaluation was conducted to scrutinize these four prominent products across diverse queries from various sources. Initially, the responses generated by current generative search engines exhibit fluency and an appearance of informativeness. However, they frequently contain unsupported assertions and inaccurately cited information. On average, only 51.5% of the generated sentences possess adequate citation support, and a mere 74.5% of the citations effectively corroborate their associated sentences. This discrepancy raises substantial concerns, especially for users who rely on these systems as their primary information-seeking resources. The existing commercial systems thus exhibit considerable room for improvement.

6.2.2 Multi-Modality Question Answering

> *The content presented within this section encompasses the material from this paper:*
>
> *Raphael Tang, *Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Jimmy Lin, and Ferhan Ture. "What the daam: Interpreting stable diffusion using cross attention." *ACL 2023* (*Equal Contribution)

The entirety of the research discussed in this thesis rests upon the conventional notion of Open-Domain Question Answering (ODQA), where the answering process is executed based on textual sources of information. Expanding on this, it's worth noting that questions can also extend to other modalities, such as images, leading to what is termed Visual Question Answering (VQA). In congruence with ODQA's

[5] Perplexity.ai

[6] https://web.youchat.com/

[7] https://neeva.com/blog/introducing-neevaai

approach of reasoning over a diverse array of documents, a novel VQA dataset called SlideVQA has been introduced (Tanaka et al., 2023). This dataset aims to address scenarios where a system, presented with a slide deck containing multiple slide images and a corresponding question, must select pertinent evidence images and provide answers.

Pushing the envelope further, the task of MULTIMODALQA (Talmor et al., 2020) involves answering questions that span across free-form text, semi-structured tables, and images. The process to tackle such questions begins with breaking them down into a sequence of simpler inquiries. Subsequently, it's crucial to determine the suitable modalities for these simplified questions and provide answers accordingly. For example, image information might be stored in visual formats, specific entity details could reside in textual content, and structured data could be embedded within tables. Ultimately, the information garnered from these simplified questions must be synthesized to formulate a comprehensive answer.

Within the realm of computer vision, endeavors have been undertaken to enhance VQA datasets, overcoming limitations posed by the scarcity and expense of image collection. A recent study by Kim et al. (2023) leverages the diffusion model (Ho et al., 2020; Sohl-Dickstein et al., 2015) to generate images with a dual purpose. The primary objective is to preserve essential answer-related information, while concurrently broadening the spectrum of image diversity. Additionally, this approach serves to infuse prior knowledge into the VQA model, enhancing its overall performance. Diverging from basic image augmentation methods like flipping and rotation, which only alter specific elements of images, the images produced by the Diffusion model exhibit varying appearances. These generated images are equipped to handle abstract descriptions present in textual prompts. Consequently, VQA models trained on such augmented datasets exhibit remarkable performance in VQA tasks.

In a recent work, we propose an attribution method to provide insight into the workings of large diffusion models. With a focus on text-to-image attribution, our central research question is, "How does an input word influence parts of a generated image?" To this, we propose to produce two-dimensional attribution

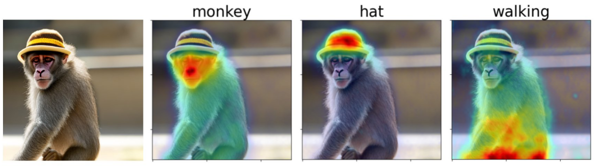


Figure 6.1: The original synthesized image and three DAAM maps for “monkey,” “hat,” and “walking,” from the prompt, “monkey with hat walking.”

maps for each word by combining cross-attention maps in the model. We name our method diffusion attentive attribution maps, or DAAM for short — see Figure 6.1 for an example. We study how relationships in the syntactic space of prompts relate to those in the pixel space of images. The findings are illustrated in Figure 6.2. We assess head–dependent DAAM interactions across ten common syntactic relations (enhanced Universal Dependencies; Schuster and Manning, 2016), finding that, for some, the heat map of the dependent strongly subsumes the head’s, while the opposite is true for others.

Finally, we form hypotheses to further our syntactic findings, studying semantic phenomena using DAAM, particularly those affecting image quality. We demonstrate that, in constructed prompts with two distinct nouns, cohyponyms have worse quality (9% worse than non-cohyponyms), e.g., “a giraffe and a zebra” generates a giraffe *or* a zebra, but not both (shown in Figure 6.3). Cohyponym status and generation incorrectness each increases the amount of heat map overlap, advancing DAAM’s utility toward improving diffusion models. We also show that descriptive adjectives attend too broadly across the image, far beyond their nouns. As shown in Figure 6.4, the DAAM map for “rusty” attends broadly, and the back-ground for “rusty” is surely not clean. When we change the adjective to “metallic” and “wooden,” the shed changes along with it, becoming grey and wooden, indicating entanglement. Similar observa- tions apply to our second case, “a bumpy, smooth, spiky ball rolling down a hill,” where “bumpy” produces rugged ground, “smooth” flatter ground, and “spiky” blades of grass. If we fix the scene layout (Hertz et al., 2022) and vary only the adjective, the entire image changes, not just the noun. These two phenomena suggest feature entanglement, where objects are entangled with both the scene and other objects. In our third case, we study color adjectives using “a {blue, green,

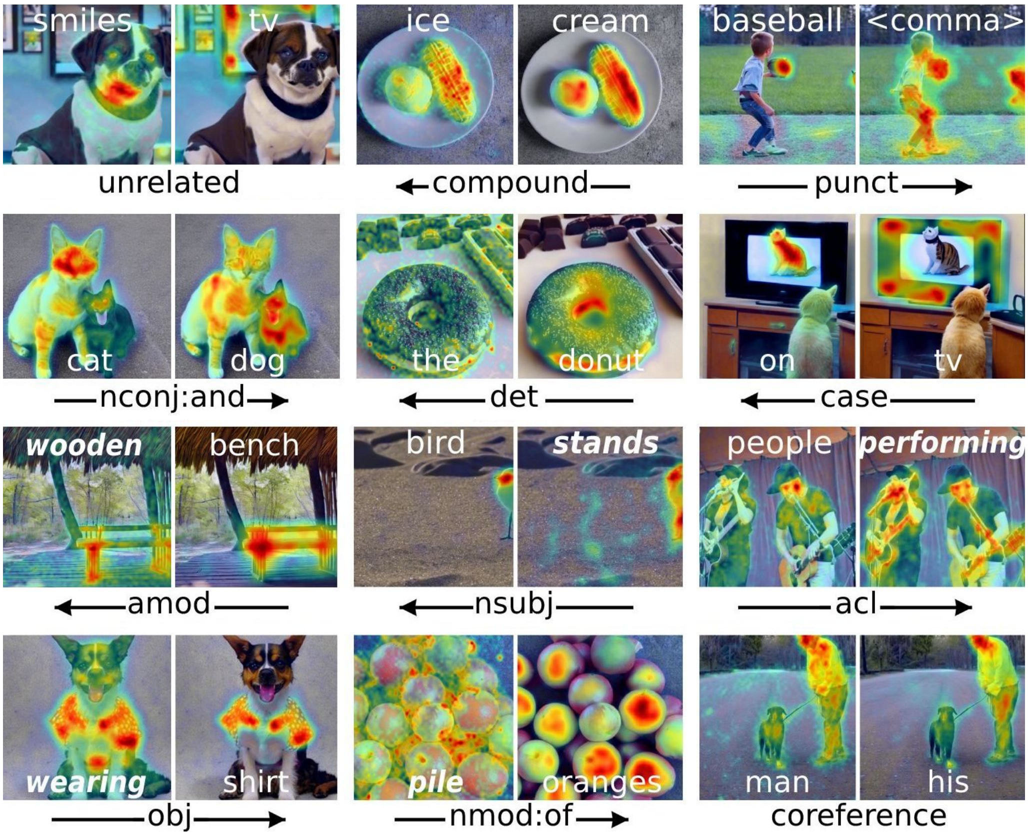


Figure 6.2: Twelve example pairs of DAAM maps, with the dominant word in bold, if present for the relation. Note that the visualization scale is normalized for each image since our purpose is to study the *spatial locality* of attribution *conditioned on the word*. For example, the absolute magnitude for the comma above is weak.

red} car driving down the streets," presented in Figure 6.5. We discover the same phenomena, with the difference that these prompts lead to *quantifiable* notions of adjectival entanglement. For, say, "green," we can conceivably measure the amount of additional green hue in the background, with the car cropped out—see bottom row. A caveat is that entanglement is not necessarily unwanted; for instance, rusty shovels likely belong in rusted areas. It strongly depends on the use case of the model.

Overall, we study visuolinguistic phenomena in diffusion models by interpreting word–pixel cross-attention maps. We prove the correctness of our attribution method, DAAM, through a quantitative semantic segmentation task and a qualitative generalized attribution study. We apply DAAM to assess how syntactic relations translate to visual interactions, finding that certain maps of heads inappropriately subsume their dependents'. We use these findings to form hypotheses about feature entanglement,

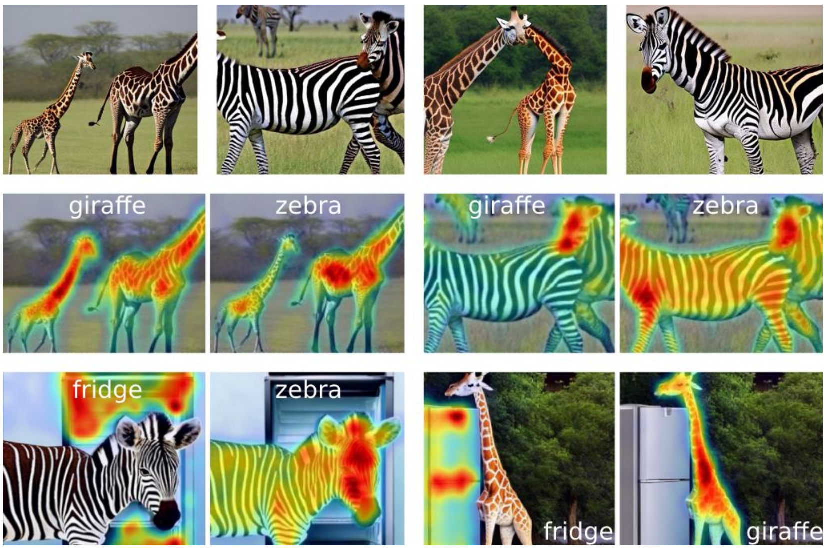


Figure 6.3: Rows starting from the top: generated images for cohyponyms “a giraffe and a zebra,” heat maps for the first two images, and heat maps for non-cohyponymic zebra–fridge and giraffe–fridge prompts.

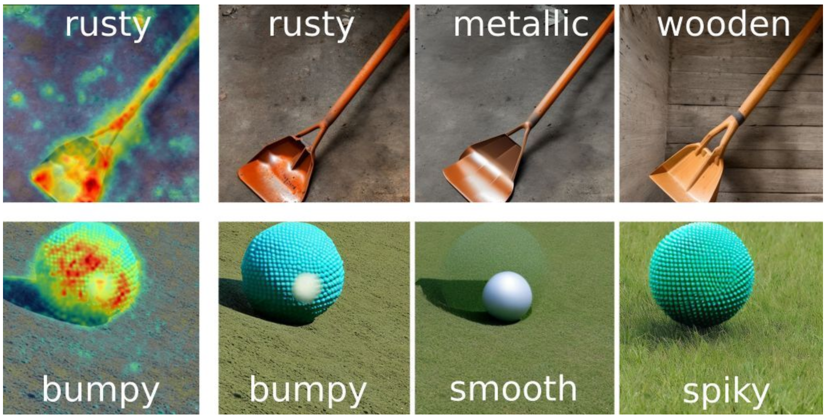


Figure 6.4: First row: a DAAM map for “rusty” and three generated images for “a `<adj>` shovel sitting in a clean shed;” second row: a map for “bumpy” and images for “a `<adj>` ball rolling down a hill.”

showing that cohyponyms are jumbled and adjectives attend too broadly. These findings have significant implications. Notably, they can contribute to the refinement of pipeline design when integrating the Diffusion model into question-answering tasks that involve multiple modalities. In essence, our research sheds light on the intricate relationship between language and visuals, offering valuable guidance for enhancing the fusion of these elements in complex tasks.

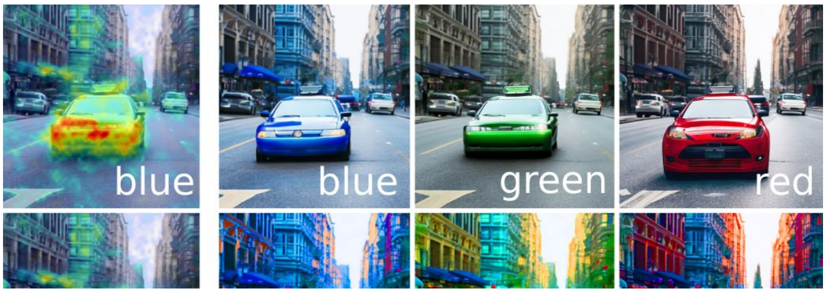
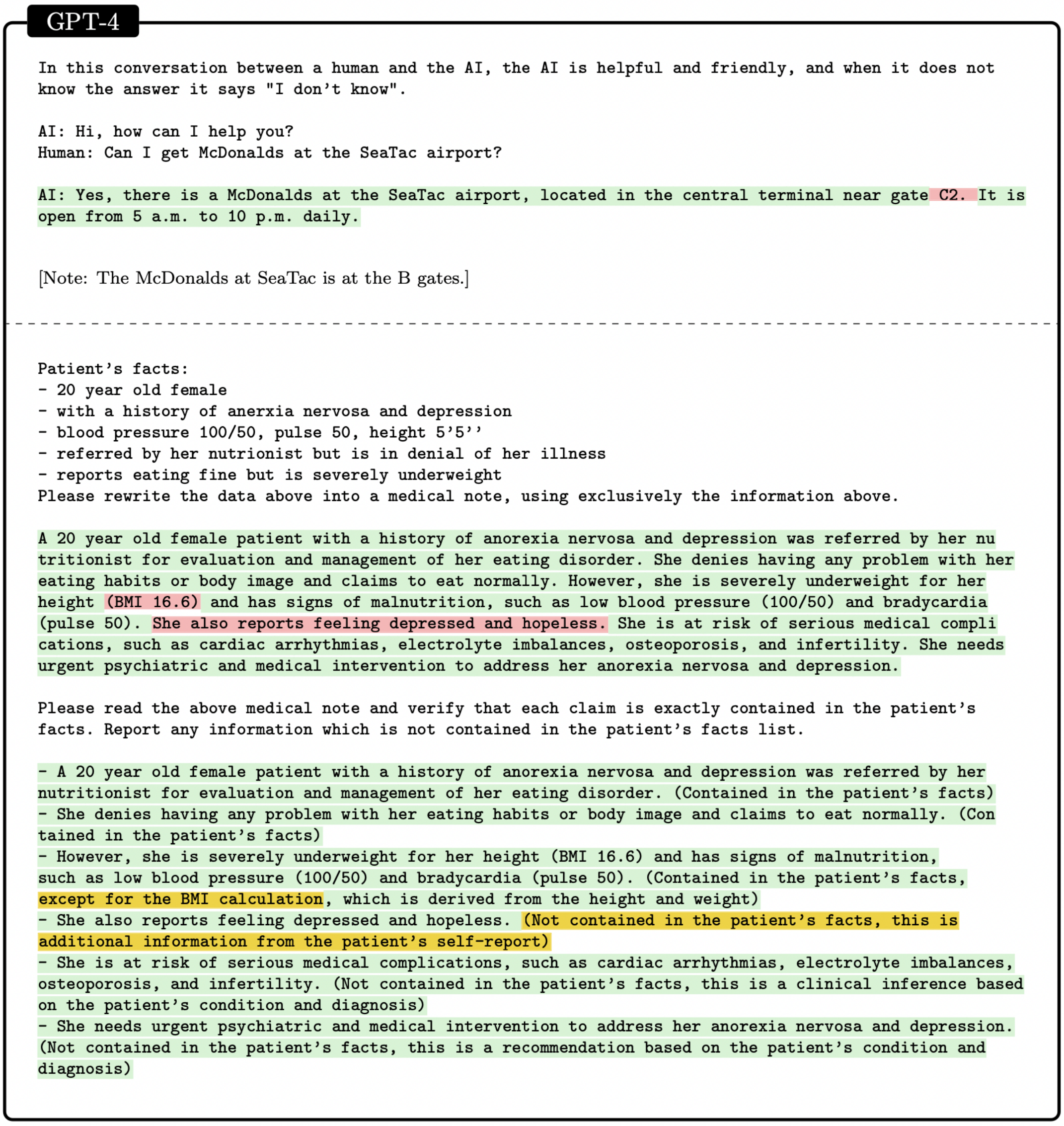


Figure 6.5: A DAAM map and generated images for "a `<adj>` car driving down the streets," above images of the cropped background, oversaturated for visualization.

6.2.3 Combating Hallucinations in Language Models

While large language models, comprising billions of parameters and trained on vast datasets, have showcased remarkable advancements across various tasks, they exhibit a susceptibility to generating text that appears linguistically coherent but deviates from the fidelity of the provided source input or strays from established factual knowledge (Bubeck et al., 2023; Vinyals and Le, 2015). This susceptibility leads to the phenomenon known as "hallucinations" (Maynez et al., 2020), wherein generated content lacks factual accuracy.

An illustrative case of both open-domain and closed-domain hallucinations involving the state-of-the-art GPT-4 model is depicted in Figure 6.6. These hallucinations can be broadly classified into two categories: 1) *Open-domain hallucinations* (shown in upper Fig. 6.6): These occur when the model produces incorrect or unsupported information in response to open-ended prompts. This can be likened to closed-book question answering, as discussed earlier in this thesis. The knowledge generated is solely derived from the model's internal parameters unless additional background research is conducted. Previous research (Kandpal et al., 2023) has demonstrated that a language model's proficiency in answering factual questions is tied to its exposure to relevant documents during pre-training. Strong correlations and causal connections have been observed between accuracy and the number of pertinent documents associated with a range of question-answering datasets, pre-training datasets, and model sizes. 2) *Closed-domain hallucinations* (shown in lower Fig. 6.6) refer to mistakes or inaccuracies that occur within a specific context, where there are existing content or constraints that can be used to validate or verify the



GPT-4

```
In this conversation between a human and the AI, the AI is helpful and friendly, and when it does not
know the answer it says "I don't know".

AI: Hi, how can I help you?
Human: Can I get McDonalds at the SeaTac airport?

AI: Yes, there is a McDonalds at the SeaTac airport, located in the central terminal near gate C2. It is
open from 5 a.m. to 10 p.m. daily.


[Note: The McDonalds at SeaTac is at the B gates.]
```

```
Patient's facts:
- 20 year old female
- with a history of anerxia nervosa and depression
- blood pressure 100/50, pulse 50, height 5'5''
- referred by her nutrionist but is in denial of her illness
- reports eating fine but is severely underweight
Please rewrite the data above into a medical note, using exclusively the information above.

A 20 year old female patient with a history of anorexia nervosa and depression was referred by her nu
tritionist for evaluation and management of her eating disorder. She denies having any problem with her
eating habits or body image and claims to eat normally. However, she is severely underweight for her
height (BMI 16.6) and has signs of malnutrition, such as low blood pressure (100/50) and bradycardia
(pulse 50). She also reports feeling depressed and hopeless. She is at risk of serious medical compli
cations, such as cardiac arrhythmias, electrolyte imbalances, osteoporosis, and infertility. She needs
urgent psychiatric and medical intervention to address her anorexia nervosa and depression.

Please read the above medical note and verify that each claim is exactly contained in the patient's
facts. Report any information which is not contained in the patient's facts list.

- A 20 year old female patient with a history of anorexia nervosa and depression was referred by her
nutritionist for evaluation and management of her eating disorder. (Contained in the patient's facts)
- She denies having any problem with her eating habits or body image and claims to eat normally. (Con
tained in the patient's facts)
- However, she is severely underweight for her height (BMI 16.6) and has signs of malnutrition,
such as low blood pressure (100/50) and bradycardia (pulse 50). (Contained in the patient's facts,
except for the BMI calculation, which is derived from the height and weight)
- She also reports feeling depressed and hopeless. (Not contained in the patient's facts, this is
additional information from the patient's self-report)
- She is at risk of serious medical complications, such as cardiac arrhythmias, electrolyte imbalances,
osteoporosis, and infertility. (Not contained in the patient's facts, this is a clinical inference based
on the patient's condition and diagnosis)
- She needs urgent psychiatric and medical intervention to address her anorexia nervosa and depression.
(Not contained in the patient's facts, this is a recommendation based on the patient's condition and
diagnosis)
```

Figure 6.6: Illustrations of hallucination occur in both open-domain and closed-domain scenarios. In the closed-domain context, it becomes evident that GPT-4 can serve as a tool for recognizing hallucinations. However, its proficiency is not flawless. For instance, GPT-4 may rationalize providing the Body Mass Index (BMI) by deducing it from height and weight, even if the weight value is not explicitly provided. The example is from Bubeck et al. (2023).

accuracy of the generated information. These hallucinations are typically errors that arise when generating content that should be consistent with the provided context, data, or guidelines. For example, consider a scenario where an AI language model (LLM) is tasked with summarizing a news article. A closed-domain hallucination might occur if the summary includes details or information that are not present in the original article or misrepresents the facts stated in the article. In this case, the context of the original article serves as a basis for verifying the accuracy of the generated summary. Addressing closed-domain hallucinations involves techniques aimed at maintaining consistency and alignment within the given context. One approach is to employ methods that check for consistency, such as using the same LLMs to identify discrepancies or fabricated information that goes beyond the provided facts or content. This can help identify instances where the AI model is generating information that is not supported by the provided data or constraints, allowing for better control over the quality and accuracy of generated content in specific contexts.

Appendix A

Appendix for Annotated Question Answer pairs based on Compositional Generalization

A.1 Answer Grounding in Retrieved Passages

We noted in Section 3.3 that we find evidence the FiD (Izacard and Grave, 2021) ODQA model does ground its answers in the retrieved passages. This observation can be contrasted to that of Krishna et al. (2021), who found that answers to long-form questions were not grounded in the passage, in that models would provide the same answer regardless of the context provided. A complete picture of the results from our experiment can be seen in Table A.1. We note that when the models is fed solely random passages it fails to answer nearly all questions (3.6%). However, but

Passage Processing	**Total**	**Overlap**	**Comp-gen**	**Novel-entity**
Original retrieved	53.1	78.9	40.0	47.7
50% random	53.2	78.3	39.9	48.3
99% random	55.5	74.3	46.1	54.0
100% random	3.6	5.1	2.0	3.0

Table A.1: Comparison of FiD's predictions for the NQ test set, conditioned on the *originally retrieved* passages and a gradually increasing number of *randomly chosen* passages. x% means the percentage of retrieved passages are replaced with random ones. For *99% random*, the rest passage is gold passage which contains the gold answer span.

Group	Test question	Train question
Overlap	Where does patience is a virtue come from	Where did the saying patience is a virtue come from
	Who was the killer in the movie I Know What You did Last Summer	Who was the murderer in I Know What You did Last Summer
	When was the last time Arsenal win Premier League	When was the last time Arsenal won the Premier League title
	Where does blood go when it leaves the pulmonary artery	Where does blood go after the pulmonary artery
Comp-gen	What is the most popular religion in Sweden	What is the most popular religion in Ukraine
	What are the main functions of the stem	What are the main functions of the control bus
	Who is in charge of ratifying treaties in the US	Who is in charge if president is impeached
	Cast of the Have and Have Nots play	The last episode of the Haves and Have Nots
Novel-entity	Where does wild caught *sockeye salmon* come from	When was *Sony walkman* first sold in stores
	The probability of making a *Type I Error* when retaining .. is	When was *tower of terror* built in Disneyland
	Who was the *Pinkerton Detective Agency* 's first female detective	Who played *detective Green* on Law & Order
	Where was the *world economic forum* held this year	Who holds the *world record* for 100 meters

Table A.2: Example questions from NQ test set.

Group	Test question	Train question
Overlap	Which is the highest waterfall in the world	What is the tallest waterfall in the world
	In the cartoon series, what kind of dog is Scooby Doo	What breed of dog is Scooby-Doo
	Who directed the film “Gladiator”, starring Russell Crowe	Who directed the film Gladiator
	Which is the largest island in Canada	What is Canada’s largest island
Comp-gen	- What nationality was the painter Vincent Van Gogh	- What nationality was painter Piet Mondrian
	- What post was held by Winston Churchill during the 1926 general strike in the UK	- What role was played by Arthur Cook In the general strike of 1926
	- By population, which is the second biggest city in France	- In terms of population, which is the second largest city in Finland 1926
	- In humans, the medical condition prepatellar bursitis affects which part of the body	- The medical condition aerotitis affects which part of the human body
Novel-entity	- In *‘follow that camel’*, the fourteenth carry on film, sid james was replaced by which us actor	- What was the cause of death of carmen in the opera *of that name*
	- Who has recently overtaken *brian o’driscoll* to become ireland’s most capped player	- In the 2005 remake of king kong, who played the writer *jack driscoll*
	- *Shining Tor* is the highest point in which county	- *Shinto* is the main religion in which country
	- Who had a *Too Legit to Quit* tour	- Which sweets were advertised as the *Too Good to Hurry Mints*

Table A.3: Example questions from TriviaQA test set.

provided with half gold and half random passages, it performs on par with its original performance. Lastly, we note that when presented with a single gold passage and otherwise only random passages, the model is still able to determine which passage is the gold passage and answer the question correctly – in fact, the performance even improves upon the original performance with more than more than 5% for comp-gen and novel-entity questions.

A.2 Additional Examples for three generalization subsets

Additional examples from Natural Questions are provided in Table A.2, WebQuestions in Table A.3, and TriviaQA datasets in Table A.4.

Group	Test question	Train question
Overlap	What is the currency of Puerto Rico called	What type of currency is used in Puerto Rico
	Which countries speak German officially	What countries speak German as a first language
	What language is spoken in Haiti today	What language do Haitian speak
	What team is Hank Baskett on 2010	What team is Hank Baskett playing for in 2010
Comp-gen	What year was George W Bush elected	What is George W Bush's middle name
	What year did the Seahawks win the Superbowl	In what Super Bowl did the Seahawks face the Steelers
	Where did Queensland get its name from	From where did the Guillotine get its name
	Where was Theodore Roosevelt buried	Where is George v1 buried
Novel-entity	Where did *Andy Murray* started playing tennis	When did *Sean Murray* first appear on NCIS
	What time in *Hilo Hawaii*	Who was *Phil Harris* married to
	Where did *Bristol Palin* go to school	What team is *Chris Paul* on
	What time does *American Horror Story* air	Who made the *American Red Cross*

Table A.4: Example questions from WebQ test set.

A.3 Influence of question pattern frequency

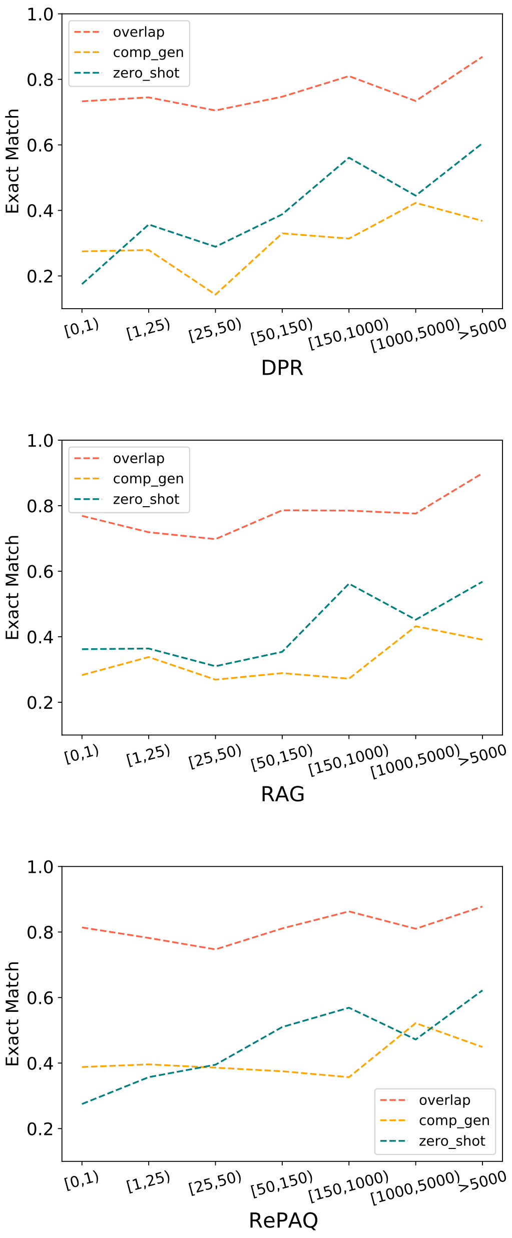


Figure A.1: Influence of question pattern frequency. Each figure is associated with one non-parametric model, which is DPR, RAG and RePAQ from left to right. The test questions are binned based on the frequency of their question pattern in the training set. The y-axis shows the Exact Match score on the NQ test set.

Bibliography

Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. Umass at trec 2004: Novelty and hard. *Computer Science Department Faculty Publication Series*, page 189, 2004.

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations*, 2020.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. Synthetic qa corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, 2019.

Ali Mohamed Nabil Allam and Mohamed Hassan Haggag. The question answering systems: A survey. *Science*, 2(3), 2012.

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average, 2019.

Avinash Atreya and Charles Elkan. Latent semantic indexing (lsi) fails for trec collections. *ACM SIGKDD Explorations Newsletter*, 12(2):5–10, 2011.

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic generalization: What is required and can it be learned? In *International Conference on Learning Representations*, 2018.

Dara Bahri, Hossein Mobahi, and Yi Tay. Sharpness-aware minimization improves language model generalization, 2021.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544, 2013.

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen-tau Yih, Sebastian Riedel, and Fabio Petroni. Autoregressive search engines: Generating substrings as document identifiers. *arXiv preprint arXiv:2204.10628*, 2022.

Devansh Bisla, Jing Wang, and Anna Choromanska. Low-pass filtering sgd for recovering flat optima in the deep learning optimization landscape, 2022. URL https://arxiv.org/abs/2201.08025.

Daniel Bobrow et al. Natural language input for a computer problem solving system. 1964.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Arbi Bouchoucha, Jing He, and Jian-Yun Nie. Diversified query expansion using conceptnet. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 1861–1864, 2013.

TL Brauen, RC Holt, and TR Wilcox. Xi. document indexing based on relevance feedback. 1968.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. SWAD: Domain generalization by seeking flat minima. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=zkHlu_3sJYU`.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=B1YfAfcgl`.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Evaluating question answering evaluation. In *Proceedings of the 2nd workshop on machine reading for question answering*, pages 119–124, 2019.

Anthony Chen, Pallavi Gudipati, Shayne Longpre, Xiao Ling, and Sameer Singh. Evaluating entity disambiguation and the role of popularity in retrieval-based nlp. *arXiv preprint arXiv:2106.06830*, 2021a.

Jifan Chen and Greg Durrett. Understanding dataset design choices for multi-hop reasoning. 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 4026–4032, 2019.

Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations, 2021b.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*, 2020.

Fabio Crestani, Mounia Lalmas, Cornelis J Van Rijsbergen, and Iain Campbell. "is this document relevant?... probably" a survey of probabilistic models in information retrieval. *ACM Computing Surveys (CSUR)*, 30(4):528–552, 1998.

W Bruce Croft and David J Harper. Using probabilistic models of document retrieval without relevance information. *Journal of documentation*, 35(4):285–295, 1979.

Zhuyun Dai and Jamie Callan. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 985–988, 2019.

Zhuyun Dai and Jamie Callan. Context-aware term weighting for first stage passage retrieval. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1533–1536, 2020.

Hoa Trang Dang, Diane Kelly, Jimmy Lin, et al. Overview of the trec 2007 question answering track. In *Trec*, volume 7, page 63, 2007.

Verna Dankers, Elia Bruni, and Dieuwke Hupkes. The paradox of the compositionality of natural language: a neural machine translation case study. *arXiv preprint arXiv:2108.05885*, 2021.

Yann N. Dauphin, Razvan Pascanu, Çaglar Gülçehre, KyungHyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2933–2941, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/17e23e50bedc63b4095e3d8204ce063b-Abstract.html.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ: hessian aware quantization of neural networks with mixed-precision. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 293–302. IEEE, 2019. doi: 10.1109/ICCV.2019.00038. URL https://doi.org/10.1109/ICCV.2019.00038.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred A. Hamprecht. Essentially no barriers in neural network energy landscape. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages

1308–1317. PMLR, 2018. URL `http://proceedings.mlr.press/v80/draxler18a.html`.

Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent Y. F. Tan. Efficient sharpness-aware minimization for improved training of neural networks, 2021.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, 2019.

Philipp Dufter, Nora Kassner, and Hinrich Schütze. Static embeddings as efficient knowledge bases? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2353–2363, 2021.

Nouha Dziri, Andrea Madotto, Osmar R Zaiane, 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, 2021.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In Gal Elidan, Kristian Kersting, and Alexander T. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017a. URL `http://auai.org/uai2017/proceedings/papers/173.pdf`.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017b.

Miles Efron, Peter Organisciak, and Katrina Fenlon. Improving retrieval of short texts through document expansion. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 911–920, 2012.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021.

M Vorheese Ellen. Overview of the trec 2001 question answering track. *Proc. 2001 Text REtrievalConference (TREC 2001)*, 2001.

Renxu Sun Jing Jiang Yee Fan, Tan Hang Cui Tat-Seng Chua, and Min-Yen Kan. Using syntactic and semantic relation analysis in question answering. In *Proceedings of the 14th Text REtrieval Conference (TREC), Gaithersburg, MD, USA*, pages 15–18. Citeseer, 2005.

Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, Jingjing Liu, and Chenguang Zhu. Accelerating real-time question answering via question generation. *arXiv preprint arXiv:2009.05167*, 2020.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, 2019.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=6Tm1mposlrM`.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant.

Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*, 2021a.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. Splade: Sparse lexical and expansion model for first stage ranking. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021b.

Stanislav Fort and Stanislaw Jastrzebski. Large scale structure of neural network loss landscapes. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6706–6714, 2019. URL `https://proceedings.neurips.cc/paper/2019/hash/48042b1dae4950fef2bd2aafa0b971a1-Abstract.html`.

Jonathan Frankle. Revisiting "qualitatively characterizing neural network optimization problems". *CoRR*, abs/2012.06898, 2020. URL `https://arxiv.org/abs/2012.06898`.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3259–3269. PMLR, 2020. URL `http://proceedings.mlr.press/v119/frankle20a.html`.

Peter I. Frazier. A tutorial on Bayesian optimization. *ArXiv*, abs/1807.02811, 2018.

Luyu Gao, Zhuyun Dai, and Jamie Callan. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3030–3042, 2021.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, 2018.

Jonas Geiping, Micah Goldblum, Phillip E. Pope, Michael Moeller, and Tom Goldstein. Stochastic training is not necessary for generalization. *CoRR*, abs/2109.14119, 2021. URL `https://arxiv.org/abs/2109.14119`.

Ian J. Goodfellow and Oriol Vinyals. Qualitatively characterizing neural network optimization problems. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6544`.

Michael Gordon. Probabilistic and genetic algorithms in document retrieval. *Communications of the ACM*, 31(10):1208–1218, 1988.

Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=r14EOsCqKX`.

Bert F Green Jr, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224, 1961.

Yashvir Grewal and Thang D Bui. Diversity is all you need to improve bayesian model averaging. In *Bayesian Deep Learning Workshop at Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, 2021.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. Beyond iid: three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, pages 3477–3488, 2021.

Hao Guo, Jiyong Jin, and Bin Liu. Stochastic weight averaging revisited. *CoRR*, abs/2201.00519, 2022. URL `https://arxiv.org/abs/2201.00519`.

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, 2018.

Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: Beyond sharp and flat local minima. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2549–2560, 2019. URL `https://proceedings.neurips.cc/paper/2019/hash/01d8bae291b1e4724443375634ccfa0e-Abstract.html`.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv:2208.01626*, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. Wikireading: A novel large-scale language understanding task over wikipedia. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1545, 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1): 1–42, 1997.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/fb60d411a5c5b72b2e7d3527cfc84fd0-Abstract.html`.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get M for free. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=BJYwwY9ll`.

Gautier Izacard and Edouard Grave. Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*, 2020.

Gautier Izacard and Édouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, 2021.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 876–885. AUAI Press, 2018. URL `http://auai.org/uai2018/proceedings/papers/313.pdf`.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, 2017.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=SJgIPJBFvH`.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Mandar Joshi, Eunsol Choi, Daniel S 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, 2017.

Jean Kaddour. Stop wasting my time! saving days of imagenet and bert training with latest weight averaging. *arXiv preprint arXiv:2209.14981*, 2022. URL `https://arxiv.org/abs/2209.14981`.

Jean Kaddour, Aengus Lynch, Qi Liu, Matt J. Kusner, and Ricardo Silva. Causal machine learning: A survey and open problems. *arXiv preprint arXiv:2206.15475*, 2022. URL `https://arxiv.org/abs/2206.15475`.

Ehsan Kamalloo, Nouha Dziri, Charles LA Clarke, and Davood Rafiei. Evaluating open-domain question answering in the era of large language models. *arXiv preprint arXiv:2305.06984*, 2023.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR, 2023.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, 2020.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, and Kentaro Inui. Realtime qa: What's the answer right now? *arXiv preprint arXiv:2207.13332*, 2022.

Nora Kassner, Benno Krojer, and Hinrich Schütze. Are pretrained language models symbolic reasoners over knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564, 2020.

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, 2018.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=H1oyRlYgg`.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*, 2019.

Kimiya Keyvan and Jimmy Xiangji Huang. How to approach ambiguous queries in conversational search: A survey of techniques, approaches, tools, and challenges. *ACM Computing Surveys*, 55(6):1–40, 2022.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015a. URL `http://arxiv.org/abs/1412.6980`.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015b.

Russell A Kirsch. Computer interpretation of english text and picture patterns. *IEEE Transactions on Electronic Computers*, (4):363–376, 1964.

Bobby Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does sgd escape local minima? In *International Conference on Machine Learning*, pages 2698–2707. PMLR, 2018.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 2019. URL `http://proceedings.mlr.press/v97/kornblith19a.html`.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. Hurdles to progress in long-form question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4940–4957, 2021.

Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 26548–26560, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/df438e5206f31600e6ae4af72f2725f1-Abstract.html`.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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, 2019a. doi: 10.1162/tacl_a_00276. URL `https://aclanthology.org/Q19-1026`.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019b.

Cody CT Kwok, Oren Etzioni, and Daniel S Weld. Scaling question answering to the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 150–161, 2001.

KL Kwok. The use of title and cited titles as document representation for automatic classification. *Information Processing & Management*, 11(8-12):201–206, 1975.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of*

Machine Learning Research, pages 5905–5914. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/kwon21b.html`.

Brenden M Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *ICML*, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6402–6413, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html`.

Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34:29348–29363, 2021.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. Learning dense representations of phrases at scale. 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 6634–6647, 2021.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, 2019.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020a.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020b.

Patrick Lewis, Barlas Oğuz, Wenhan Xiong, Fabio Petroni, Wen-tau Yih, and Sebastian Riedel. Boosted dense retriever. *arXiv preprint arXiv:2112.07771*, 2021a.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, 2021b.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115, 2021c.

Belinda Z Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. Efficient one-pass end-to-end entity linking for questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6433–6441, 2020a.

Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. Parade: Passage representation aggregation for document reranking. *ACM Transactions on Information Systems*, 2020b.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6391–6401, 2018. URL `https://proceedings.neurips.cc/paper/2018/hash/a41b3bb3e6b050b6c9067c67f663b915-Abstract.html`.

Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Jimmy Lin. A proposed conceptual framework for a representational approach to information retrieval. In *ACM SIGIR Forum*, volume 55, pages 1–29. ACM New York, NY, USA, 2022.

Jimmy Lin and Xueguang Ma. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. *arXiv preprint arXiv:2106.14807*, 2021a.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2356–2362, 2021.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. *Pretrained transformers for text ranking: Bert and beyond.* Springer Nature, 2022.

Jimmy J. Lin and Xueguang Ma. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. *ArXiv*, abs/2106.14807, 2021b.

Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, D'Autume Cyprien De Masson, Tim Scholtes, Manzil Zaheer, Susannah Young, et al. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*, pages 13604–13622. PMLR, 2022.

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. *arXiv preprint arXiv:2307.03172*, 2023a.

Nelson F Liu, Tianyi Zhang, and Percy Liang. Evaluating verifiability in generative search engines. *arXiv preprint arXiv:2304.09848*, 2023b.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, 2021.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. Sparse, Dense, and Attentional Representations for Text Retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345, 04 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00369. URL `https://doi.org/10.1162/tacl_a_00369`.

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 *Proceedings of the 1st Workshop on Semiparametric Methods in NLP: Decoupling Logic from Knowledge*, pages 7–22, 2022.

Xueguang Ma, Minghan Li, Kai Sun, Ji Xin, and Jimmy Lin. Simple and effective unsupervised redundancy elimination to compress dense vectors for passage retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2854–2859, Online and Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics.

Xueguang Ma, Kai Sun, Ronak Pradeep, and Jimmy Lin. A replication study of dense passage retriever. *arXiv preprint arXiv:2104.05740*, 2021b.

Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonellotto. Learning passage impacts for inverted indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1723–1727, 2021.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, 2021.

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, 2020.

Yusuf Mehdi. The new bing and edge – progress from our first month, 2023. URL `https://blogs.bing.com/search/march_2023/The-New-Bing-and-Edge-%E2%80%93-Momentum-from-Our-First-Month/`.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. Ambigqa: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Con-*

ference on Empirical Methods in Natural Language Processing (EMNLP), pages 5783–5797, 2020.

RV Mises and Hilda Pollaczek-Geiringer. Praktische verfahren der gleichungsauflösung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 9(1):58–77, 1929.

Diego Mollá, Menno Van Zaanen, and Daniel Smith. Named entity recognition for question answering. In *Proceedings of the Australasian language technology workshop 2006*, pages 51–58, 2006.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906, 2018.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/0607f4c705595b911a4f3e7a127b44e0-Abstract.html`.

Evgenii Nikishin, Pavel Izmailov, Ben Athiwaratkun, Dmitrii Podoprikhin, Timur Garipov, Pavel Shvechikov, Dmitry Vetrov, and Andrew Gordon Wilson. Improving stability in deep reinforcement learning with weight averaging. In *Uncertainty in artificial intelligence workshop on uncertainty in Deep learning*, 2018.

Rodrigo Nogueira and Kyunghyun Cho. Task-oriented query reformulation with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 574–583, 2017.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*, 2019a.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*, 2019b.

Maxwell Nye, Michael Tessler, Josh Tenenbaum, and Brenden M Lake. Improving coherence and consistency in neural sequence models with dual-system, neuro-symbolic reasoning. *Advances in Neural Information Processing Systems*, 34, 2021.

Barlas Oguz, Kushal Lakhotia, Anchit Gupta, Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Scott Yih, Sonal Gupta, et al. Domain-matched pre-training tasks for dense retrieval. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1524–1534, 2022.

OpenAI. Introducing gpt-4. `https://openai.com/blog/gpt-4/`, 2023. Accessed: [insert date here].

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

María Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári. Tighter risk certificates for neural networks. *Journal of Machine Learning Research*, 22(227):1–40, 2021.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, 2019.

Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. Relative flatness and generalization. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=sygvo7ctb_`.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual meeting of the association for Computational Linguistics*, pages 41–47, 2002.

Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. Pair: Leveraging passage-centric similarity relation for improving dense passage retrieval. *arXiv preprint arXiv:2108.06027*, 2021.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, 2020.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

Stephen E Robertson and K Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146, 1976.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at trec-3, trec'94: Proceedings of the 3rd text retrieval conference, gaithersburg, ma 1994, ed. by donna k. *Harman (NIST, Gaithersburg, MA 1994)*, pages 109–126.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *arXiv preprint arXiv:2107.12708*, 2021.

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M Lake. A benchmark for systematic generalization in grounded language understanding. *Advances in Neural Information Processing Systems*, 33, 2020.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Representations by Back-Propagating Errors*, page 696–699. MIT Press, Cambridge, MA, USA, 1988. ISBN 0262010976.

Gerard Salton. *The SMART retrieval system—experiments in automatic document processing*. Prentice-Hall, Inc., 1971.

Gerard Salton. A new comparison between conventional indexing (medlars) and automatic text processing (smart). *Journal of the American Society for Information Science*, 23(2):75–84, 1972.

Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

Adepu Ravi Sankar, Yash Khasbage, Rahul Vigneswaran, and Vineeth N Balasubramanian. A deeper look at the hessian eigenspectrum of deep neural networks and its applications to regularization. *arXiv preprint arXiv:2012.03801*, 2020.

Sebastian Schuster and Christopher D. Manning. Enhanced English universal dependencies: An improved representation for natural language understanding tasks. In

Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), 2016.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*, 2019.

Robert F Simmons. Answering english questions by computer: a survey. *Communications of the ACM*, 8(1):53–70, 1965.

Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. End-to-end training of multi-document reader and retriever for open-domain question answering. *Advances in Neural Information Processing Systems*, 34:25968–25981, 2021.

Amit Singhal and Fernando Pereira. Document expansion for speech retrieval. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 34–41, 1999.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*, 2022.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

Luca Soldaini and Alessandro Moschitti. The cascade transformer: an application for efficient answer sentence selection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5697–5708, 2020.

Martin M Soubbotin and Sergei M Soubbotin. Patterns of potential answer expressions as clues to the right answers. In *TREC*, 2001.

David Stutz, Matthias Hein, and Bernt Schiele. Relating adversarially robust generalization to flat minima. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 7787–7797. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00771. URL `https://doi.org/10.1109/ICCV48922.2021.00771`.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. What makes reading comprehension questions easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, 2018.

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. Multimodalqa: complex question answering over text, tables and images. In *International Conference on Learning Representations*, 2020.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. Slidevqa: A dataset for document visual question answering on multiple images. *arXiv preprint arXiv:2301.04883*, 2023.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P Parikh. Sticking to the facts: Confident decoding for faithful data-to-text generation. *arXiv preprint arXiv:1910.08684*, 2019.

Andrew Trotman. Learning to rank. *Information Retrieval*, 8:359–381, 2005.

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, 2017a.

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, 2017b.

Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

Ellen Voorhees. Overview of the trec 2004 question answering track. 01 2004.

Ellen Voorhees and Hoa Dang. Overview of the trec 2005 question answering track. 01 2005.

Ellen M Voorhees. Question answering in trec. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 535–537, 2001.

Ellen M. Voorhees. Overview of the trec 2002 question answering track. In *Text Retrieval Conference*, 2003. URL `https://api.semanticscholar.org/CorpusID:215762892`.

Ellen M Voorhees and L Buckland. Overview of the trec 2003 question answering track. In *TREC*, volume 2003, pages 54–68, 2003.

Ellen M Voorhees et al. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82, 1999.

Shuai Wang, Shengyao Zhuang, and Guido Zuccon. Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 317–324, 2021.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Xing Wei and W Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, 2006.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020.

Yihan Wu, Aleksandar Bojchevski, and Heng Huang. Adversarial weight perturbation improves generalization in graph neural networks, 2022. URL `https://openreview.net/forum?id=hUr6K4D9f7P`.

Yuxiang Wu, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Don't read too much into it: Adaptive computation for open-domain question answering. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 63–72, 2020.

Jinfeng Xiao, Lidan Wang, Franck Dernoncourt, Trung Bui, Tong Sun, and Jiawei Han. Open-domain question answering with pre-constructed question spaces. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 61–67, 2021.

Chenyan Xiong and Jamie Callan. Query expansion with freebase. In *Proceedings of the 2015 international conference on the theory of information retrieval*, pages 111–120, 2015.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2020.

Jinxi Xu and W Bruce Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems (TOIS)*, 18 (1):79–112, 2000.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=ryGs6iA5Km`.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. Efficient passage retrieval with hashing for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 979–986, 2021.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, 2019.

Yaoqing Yang, Liam Hodgkinson, Ryan Theisen, Joe Zou, Joseph E. Gonzalez, Kannan Ramchandran, and Michael W. Mahoney. Taxonomizing local versus global structure in neural network loss landscapes. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=P6bUrLREcne`.

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, 2015.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael Mahoney. LARGE BATCH SIZE TRAINING OF NEURAL NETWORKS WITH ADVERSAR-

IAL TRAINING AND SECOND-ORDER INFORMATION, 2019. URL https://openreview.net/forum?id=H1lnJ2Rqt7.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W. Mahoney. Pyhessian: Neural networks through the lens of the hessian. In Xintao Wu, Chris Jermaine, Li Xiong, Xiaohua Hu, Olivera Kotevska, Siyuan Lu, Weija Xu, Srinivas Aluru, Chengxiang Zhai, Eyhab Al-Masri, Zhiyuan Chen, and Jeff Saltz, editors, *2020 IEEE International Conference on Big Data (IEEE BigData 2020), Atlanta, GA, USA, December 10-13, 2020*, pages 581–590. IEEE, 2020. doi: 10.1109/BigData50022.2020.9378171. URL https://doi.org/10.1109/BigData50022.2020.9378171.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. Pretrained transformers for text ranking: BERT and beyond. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, pages 1–4, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-tutorials.1. URL https://aclanthology.org/2021.naacl-tutorials.1.

Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. Applying bert to document retrieval with birch. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 19–24, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. BMVA Press, 2016. URL http://www.bmva.org/bmvc/2016/papers/paper087/index.html.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen.

Adversarial retriever-ranker for dense text retrieval. In *International Conference on Learning Representations*, 2021.

Michael Zhang and Eunsol Choi. Situatedqa: Incorporating extra-linguistic contexts into qa. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, 2021.

Yang Zhao, Hao Zhang, and Xiuyuan Hu. Ss-sam: Stochastic scheduled sharpness-aware minimization for efficiently training deep neural networks. *arXiv preprint arXiv:2203.09962*, 2022.

Pan Zhou, Jiashi Feng, Chao Ma, Caiming Xiong, Steven Chu Hong Hoi, et al. Towards theoretically understanding why sgd generalizes better than adam in deep learning. *Advances in Neural Information Processing Systems*, 33:21285–21296, 2020.

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. *arXiv preprint arXiv:2101.00774*, 2021.

Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha C Dvornek, sekhar tatikonda, James s Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=edONMAnhLu-`.