

Neural Retriever-Reader for Information Retrieval and Question Answering

by

Man Luo

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2023 by the
Graduate Supervisory Committee:

Chitta Baral, Chair
Yezhou Yang
Eduardo Blanco
Danqi Chen

ARIZONA STATE UNIVERSITY

May 2023

ABSTRACT

In the era of information explosion and multi-modal data, information retrieval (IR) and question answering (QA) systems have become essential in daily human activities. IR systems aim to find relevant information in response to user queries, while QA systems provide concise and accurate answers to user questions. IR and QA are two of the most crucial challenges in the realm of Artificial Intelligence (AI), with wide-ranging real-world applications such as search engines and dialogue systems. This dissertation investigates and develops novel models and training objectives to enhance current retrieval systems in textual and multi-modal contexts. Moreover, it examines QA systems, emphasizing generalization and robustness, and creates new benchmarks to promote their progress.

Neural retrievers have surfaced as a viable solution, capable of surpassing the constraints of traditional term-matching search algorithms. This dissertation presents Poly-DPR, an innovative multi-vector model architecture that manages test-query, and ReViz, a comprehensive multimodal model to tackle multi-modality queries. By utilizing IR-focused pretraining tasks and producing large-scale training data, the proposed methodology substantially improves the abilities of existing neural retrievers. Concurrently, this dissertation investigates the realm of QA systems, referred to as “readers”, by performing an exhaustive analysis of current extractive and generative readers, which results in a reliable guidance for selecting readers for downstream applications. Additionally, an original reader (Two-in-One) is designed to effectively choose the pertinent passages and sentences from a pool of candidates for multi-hop reasoning. This dissertation also acknowledges the significance of logical reasoning in real-world applications and has developed a comprehensive testbed, LogiGLUE, to further the advancement of reasoning capabilities in QA systems.

ACKNOWLEDGMENTS

This thesis is a collection of the collective wisdom and heartfelt contributions from various individuals. I am immensely thankful to my Ph.D. advisor, Dr. Chitta Baral, who consistently offers insightful advice in the most concise and clear manner. He has granted me the freedom to explore research problems and has guided me in shaping impactful research ideas. Beyond academia, Dr. Chitta Baral has offered personal support and valuable advice for shaping my career trajectory. I also extend my gratitude to my committee members, Dr. Yezhou Yang, Dr. Eduardo Blanco, and Dr. Danqi Chen, for their astute feedback on my research.

I am thankful to my loving parents and incredible siblings, whose unconditional love and support have guided me through moments of self-doubt. They have always believed in me, even when I lost confidence in myself. When I first traveled 7,000 miles from China to America, I never imagined that it would result in a five-year separation from them. However, their unwavering care has never made me feel as if I left home or was alone.

I would like to extend special thanks to my friends who are like family - Jacob, Maggie, Mihir, Mirali, and Saven - for taking care of me during times of poor health. There is an old saying that a true friend is someone who stands by you during difficult times. I am fortunate to have you all by my side.

I am grateful to all of my exceptional collaborators, especially Tejas, Swaroop, Arindam, Mihir, Ming, and Neeraj, for their inspiring research discussions. I also want to thank my friends Guan Lin, Jinbing Huang, Lu Cheng, and Jinyung Hong, for making my time at ASU as a Ph.D. student so enjoyable.

During my Ph.D., I completed three amazing internships in the industry, working with my outstanding mentors Kazuma Hashimoto and Yingbo Zhou at Salesforce, Shashank Jain at Meta, and Xin Xu, Vincent Zhao, and Zhuyun Dai at Google. I also

thank my peers for engaging in insightful discussions and my intern friends Anothey Chen, Akshita Jha, Albert Webson, Jinhyuk Lee, Rimita Lahiri, Zhiwei Liu, Xin Ye, and Hao Peng.

Last but not least, I would like to express my appreciation to my friend Shuguang Chen, who played a crucial role in influencing my decision to pursue a Ph.D. in 2018. Additionally, I am grateful to Dr. Joohyung Lee, who guided my initial research endeavors.

If someone tells you that Ph.D. is easy, I am 100 percent sure they are lying. If someone tells you that they are grateful for the Ph.D. journey, I am 200 percent sure that they are honest. After 5-year pursuit of a Ph.D. at ASU, I have emerged as a more resilient individual, capable of addressing complex research issues and ready to overcome any hurdles life might throw my way.

TABLE OF CONTENTS

	Page
LIST OF TABLES	x
LIST OF FIGURES	xvi
CHAPTER	
1 INTRODUCTION	1
1.1 Overview for Information Retrieval and Question Answering	1
1.2 Thesis Outline and Contribution	4
2 TEXT RETRIEVER: IMPROVING BIOMEDICAL INFORMATION RETRIEVAL WITH NEURAL RETRIEVERS	8
2.1 Overview of Existing Retriever.....	11
2.1.1 Bag-of-Words Retriever: BM25	13
2.1.2 Neural Retriever: Dense Passage Representation	14
2.1.3 Neural Retriever: Poly-Encoder	14
2.2 Poly-DPR: Multi Vectors Dense Passage Representation	15
2.3 Pretraining Tasks For Neural Retriever	16
2.3.1 Expanded Title Mapping Pretraining Task.	17
2.3.2 Reduced Sentence Mapping Pretraining Task.	18
2.4 Question Generation For Large Scale Training	18
2.4.1 Template Based Question Generation: TempQG	19
2.5 Experiments and Results	22
2.5.1 Results.....	23
2.5.2 Error Analysis	28
2.6 Discussion and Summary	30
3 LIGHT HYBRID RETRIEVER FOR EFFICIENCY AND GENERAL- IZATION	31

CHAPTER	Page
3.1 Related Work	33
3.2 Model	35
3.2.1 Preliminary	35
3.2.2 LITE: A Light Dense Retriever	37
3.2.3 Memory Efficient Hybrid Model	38
3.3 Datasets to Probe Robustness of IR Systems.....	39
3.4 Experiments and Results	40
3.4.1 Memory Efficiency and Performance.....	41
3.4.2 Generalization Results	43
3.4.3 Robustness Results.....	45
3.4.4 Ablation Study	46
3.5 Discussion and Summary	48
4 NEURAL-RERANKER	50
4.1 Related Work	52
4.2 Negative Candidate Scoring Approach.....	54
4.2.1 Training an STS-model	55
4.2.2 Negativeness Score Generation	55
4.3 Experiments and Results	56
4.3.1 Dataset and Evaluation Metric.....	56
4.3.2 Baselines	57
4.3.3 Results and Analysis	58
4.3.4 Ablation Study	60
4.4 Discussion and Summary	62

CHAPTER	Page
5 MULTIMODAL RETRIEVER FOR KNOWLEDGE BASED VISUAL QUESTION ANSWERING	67
5.1 An Overview of Knowledge-Based Question Answering	68
5.2 A Knowledge Collection Approach	69
5.3 Multi-modal Retriever	71
5.3.1 Term-based Retriever.....	71
5.3.2 Multimodal Neural Retriever.....	72
5.4 Experiments and Results	73
5.5 Dissussion and Summary	75
6 END-TO-END MULTIMODAL RETRIEVER: REVIZ	77
6.1 Related Work	80
6.2 Retrieval with Multimodal Queries	81
6.2.1 Problem Statement	81
6.2.2 ReMuQ Dataset Creation.....	81
6.3 ReViz Model	83
6.3.1 Model Architecture	84
6.3.2 Training.....	85
6.4 Pretraining Task for VL Retriever.....	86
6.5 Experiments and Results	87
6.5.1 Zero-shot Retrieval.....	88
6.5.2 Fine-tuning Performance	91
6.5.3 Compare ReViz with Existing Methods	92
6.5.4 Effects of Mask Ratio in VL-ICT Task.....	93
6.5.5 Effect of Generated Captions.....	93

CHAPTER	Page
6.6 Discussion and Summary	94
7 READING COMPREHENSION MODELS FOR TEXT	96
7.1 Extractive Reader	96
7.2 Generative Reader	97
7.3 Compare Extractive and Generative Reader	98
7.3.1 Motivation	98
7.3.2 Experiments and Results	99
7.3.3 Analysis.....	101
7.4 Discussion and Summary	104
8 GENERALIZATION AND ROBUSTNESS OF READER	106
8.1 Categorization of Domain Generalization Methods	108
8.2 Experiments and Results	111
8.3 Discussion and Summary	113
9 TEXT READER: SELECT BEFORE YOU ANSWER	115
9.1 Method	117
9.1.1 Model: Two-in-One Framework	117
9.1.2 Consistency Constraint	119
9.1.3 Similarity Constraint	119
9.2 Experiment and Results	120
9.3 Discussion and Summary	123
10 LOGICAL REASONING OF READER	125
10.1 Logical Reasoning Benchmark: LogiGLUE.....	126
10.2 Classification Models For Logical Reasoning Tasks	128
10.3 LogiT5: A generative Model for Logical Reasoning Tasks	130

CHAPTER	Page
10.4 Experiments and Result	130
10.4.1 Performance of Classification Models	130
10.4.2 Performance of a Generative Model LogiT5	131
10.4.3 Logical Reasoning and Commonsense Reasoning	131
10.4.4 GPT-3 Performance	133
10.5 Discussion and Summary	133
11 MULTIMODAL READER: READING COMPREHENSION FROM TEXT AND IMAGE.....	135
11.1 Visual Question Answering	135
11.1.1 Existing Reader.....	135
11.1.2 Proposed Reader.....	135
11.2 Experiments and Results	136
11.3 Discussion and Summary	138
12 EVALUATION FOR VQA	140
12.1 Existing Evaluation	140
12.2 Semantic Evaluation: Alternative Answer Set	141
12.3 Experiments and Results	143
12.3.1 Baseline Methods	144
12.3.2 Training with AAS.....	144
12.3.3 Evaluation of AAS	145
12.4 Discussion and Summary	146
13 CONCLUSION	147
13.1 Future Work	147
13.1.1 A Biomedical Dialogue Agent Using IR and QA	148

CHAPTER	Page
13.1.2 Retrieval-augmented LM to Address Hallucination	148
13.1.3 Efficient and Small Language Model	149
REFERENCES	150

LIST OF TABLES

Table	Page
2.1 Illustrative Examples from the BioASQ Challenge Along with the Context Retrieved by Two Methods BM25 and DPR.	10
2.2 Illustrative Examples for Templates and Questions Generated by TempQG	21
2.3 Effect of Pre-Training Tasks (PT) and Fine-Tuning Datasets (B: BioASQ, T: TempQG and A: AnsQG) on the Performance of Poly-DPR with Two Context Lengths (CL) on the BioASQ Small Corpus Test Set. Bi Stands for the i^h Batch in the Testing Sets.	23
2.4 Comparison Between Our Poly-DPR (P-DPR) with Baseline Methods in the Small Corpus and Large Corpus Settings. The Bottom Section Shows Performance of Existing Methods that Make Improvements in the Re-Ranking Method.	25
2.5 Two Best NR Models in Short and Long Context: the First Block Is Poly-DPR Pretrained with RSM and Fine-Tuned on TempQG (Short); the Second Block Is Poly-DPR Pretrained with ETM and Fine-Tuned on TempQG (Long).	26
2.6 Effect of Number of Templates (NT) on Performance.	26
2.7 Comparison Among Different Values of K for Poly-DPR in Both Short and Long Context Settings.	27
2.8 Examples of the Common Failure Modes of BM25 and Poly-DPR.	29
3.1 Performance of Existing Methods, Our Baselines and Our Hybrid Model on NQ Dataset. The Performance of DrBoost on NQ Is Using 6 Weak Learners (15.4 GB Indexing Memory) and of EntityQuestion Is Using 5 Weak Learners (13.5 GB).	42

Table	Page
3.2 Performance of Light Retrievers on BEIR in Terms of NDCG@10. MS MARCO Is Evaluated on the Dev Set. Hybrid-2: Hybrid-DrBoost-2, Hybrid-L: Hybrid-LITE.....	44
3.3 Ori: Original Question; CS: CharSwap; WD:Word Deletion; WSR: WordNet Synonym Replacement; WOR: Word Order Swaps; RSI :Random Synonym Insertion; BT: Back Translation. The Smaller the Average Drop Is, the More Robust the Model Is.	46
3.4 Three DrBoost (with 2 Weak Learners) and One Hybrid Retriever. O-DrBoost: the Original DrBoost, R-DrBoost:replace the First Weak Learner in O-DrBoost with LITE, LITE-DrBoost: Use LITE as the First Weak Learner and Mine Negative Using LITE to Train a New Weak Learner to Form a DrBoost, H-LITE-DrBoost: Hybrid BM25 with LITE-DrBoost.	47
3.5 Compare Three Hybrid Scores. We Study Two Hybrid Model, BM25 with 2 Weak Learners (32*2) and BM25 with 6 Weak Learners (32*6) .	49
4.1 Two Examples from the STS-Benchmark, the First Pair of Sentences Have the Highest Score Since They Are Highly Similar, while the Second Pair Has the Lowest Score because They Have Totally Different Meanings.	54
4.2 Bold Number Means the Best Performance in the Column of Each Block. SCONER Outperforms All Baselines. Generating Negativeness Score Using Data Augmentation Is Important to Yield Good Performance.	64

Table	Page
4.3 We Initialize Each Model Using the STS Model. We Use Green Color to Indicat Increasement Compared to the Corresponding Result of Vanilla RoBERTa, and Red for a Decrease. In Most Cases, the STS Model Is Better Than RoBERTa.	65
4.4 Each Model Is Initialized with RoBERTa Model. Three SCONER Using Generated Scores Beat Best Baseline. Using Score 5 Is Better Than Generated Scores. ↓ Means Decrease Compared to Fix Score.	65
4.5 # Means the Number of Re-Ranking Candidates and HQA Means HotpotQA Dataset. When the Recall of a Small Size Candidate Is High (E.g. 99%), Using Small-Size of Candidates in Re-Ranking Is Better. . .	66
5.1 Evaluation of Three Proposed Visual Retrievers on Precision and Recall: Caption-DPR Achieves the Highest Precision and Recall on All Number of Retrieved Knowledge.	74
5.2 Recall Increases when the Caption-DPR Method Retrieves Knowledge from a Complete Knowledge Corpus Created Using Train and Test Questions.	75
6.1 Zero-Shot Performance of ReViz and Baselines on Two Datasets: OKVQA and ReMuQ. OKVQA Is Evaluated on Two Knowledge Sources. ReViz Shows Superior Zero-Shot Performance in Majority of the Cases.	88
6.2 Comparison of ReViz when It Is Fine-Tuned on Downstream Tasks. We Compare ReViz and ReViz+VL-ICT (Our Pretraining Task). VL-ICT Enables ReViz to Be a Stronger Multimodal-Query Retrieval Model....	89

Table	Page
6.3 Comparison of Our Best Model with Existing Models on OKVQA. “FT” Denotes Fine-Tuning. Our Model Surpasses Existing Methods by Significant Margins with or Without Fine-Tuning and with Different Knowledge Corpus.	91
7.1 Comparison of Readers Based on the Different PrLMs by F1 Score. Inference Length of T5 Is Full Length of Context, 512 for ELECTRA, and 1024 for BART and RoBERTa. TQA: TriviaQA; SQA: SearchQA; HQA: HotpotQA; NQ: NaturalQuestions; TbQA: TextbookQA; RE: RelationExtraction. Bold Numbers Denote for the Best Result and Underline Numbers for the Second Best.	99
7.2 Compare Extractive and Generative Readers in Terms of Rare and Normal Answers. Ro for RoBERTa and EL for ELECTRA.	104
7.3 Examples of Questions with Answers Containing Rare Characters and the Prediction of T5-Gen.	104
8.1 QA Result: Source (IID) Accuracy and Domain Generalization (OOD) on the Question Answering Benchmark with NaturalQuestions as Source Dataset. EM: Exact-Match.	111
8.2 QA Result: Comparison of Robustness in Terms of Model-Based Eval- uation (Number of Queries Needed to Fool the Model) and Model-Free (Accuracy on Adversarial Transformations).	111

Table	Page
9.1 The Results for Two Baselines and Two-In-One Model with Similarity Constraint on Dev Set of HotpotQA Distracting Dataset. SP Stands for Supporting Facts and EM for Exact Match. * Refers to Estimation. The Bottom Systems Have Much Larger Model Size Than Our Method, where QUARK, Is the Result of a Framework with 3 BERT Models, SAE Uses Two Large Language Models and an GNN Model, and HGN Uses a Large Language Model, a GNN Model and Other Reasoning Layers.	120
9.2 The Results for Two-In-One Model with or Without Consistency and Similarity Constraints.	123
10.1 Statistics of In-Domain (IID) and Out-Of-Domain (OOD) Datasets of LogiGLUE Benchmark.	127
10.2 Results of Single Classification (SG), Multiple-Choice Question Answering (MCQA), and Answer Extraction (EXT) Models on the LogiGLUE Benchmark. The Best Performance Is Highlighted in Bold, and the Second Best Is in Underline. * Denotes F1 Score.	129
10.3 LogiT5 Achieves Better Performance Than T5, Demonstrating the Benefits of Training on a Collection of Logical Reasoning Tasks.	131
10.4 Results of Transfer Unicorn Model, a Model with Commonsense Reasoning Capacity, to LogiGLUE Benchmark. The Commonsense Reasoning Is Beneficial for the Logical Reasoning.	132
10.5 Results of Transfer LogiT5 Model, a Model Trained on LogiGLUE Benchmark, to Rainbow Benchmark. The Logical Reasoning Shows Little Benefit for the Commonsense Reasoning.	132

Table	Page
10.6 Results for Prompt Learning on LogiGLUE Using GPT-3.	133
11.1 Performance on the OK-VQA Test-Split. Our Model Outperforms Existing Methods. † Means Given Oracle Knowledge to the Reader. GS-Google Search (Training Corpus). W-Wikipedia, C-ConceptNet, GI-Google Image, Acc-Accuracy.	139
12.1 Six Types of Issues Observed in the GQA Dataset, Their Definition and Their Distribution Observed in Manual Review of 600 Samples from Testdev Balanced Split.	141
12.2 The Evaluation of Two Models on GQA and VQA with Original Metric and AAS Based Metrics.	144
12.3 Incorporate AAS in the Training Phase of LXMERT ($LMXERT_{AAS}$) on GQA Dataset.	145
12.4 The IoU Scores Between Human Annotations and AAS Based on Five Approaches.	146

LIST OF FIGURES

Figure	Page
2.1 Architectures of Three Major Types of Retrievers. For Simplicity, Some Lines in the Figures Are Not Drawn. Blue Blocks Represent the Encoding for Question, and the Green Blocks Represent Context or Documents.	12
2.2 Overview of Template-Based Question Generation.	19
2.3 Poly-DPR Is Pre-Trained on Two Novel Tasks Designed Specifically for Information Retrieval Applications. This Figure Illustrates the Sample Generation Pipeline Using the Title and Abstract from Each Sample in BioASQ.	19
3.1 The Teacher Model (DrBoost) Consists of N Weak-Learners and Produces Embeddings of Dimension $N*D$. The Student Model (LITE) Has One Weak-Learner and Produces Two Embeddings: One Has Dimension of D , and One Has Dimension of $N*D$. The Smaller Embeddings Learn to Maximize the Similarity Between Question and Positive Context Embeddings, and the Larger Embeddings Learn the Embeddings from the Teacher Model.	32
3.2 Examples of the Adversarial Attack Questions. Underline Denotes the Change from the Original Question. The Example from the Top to the Bottom Are Augmented by CS, WD, SR, WOS, SI, and BT.	40
3.3 Compare DrBoost, BM25 and the Hybrid Models Performance.	48
4.1 An Example from the HotpotQA Dataset. While Both S1 and S2 Are Negative Candidates to the Question, Our Approach Assigns a Higher Negativeness Score to S1 Than S2.	51

Figure	Page
4.2 (A) Training Pipeline: Step1--Retrieve Negative Candidates for a Question Using BM25; Step2--Use a Frozen STS Model to Generate Negativeness Scores for a Question and Candidate Pair; and Step3--Train a Neural Re-Ranker Using the Generated Scores Given by the STS Model. (b) Inference Pipeline: Retrieve the Top 100 Candidates Using BM25 and Re-Rank Them Using a Neural Re-Ranker. Q' and A' Means Augmented Questions and Answers, and S' Means Predicted Scores of Neural Re-Ranker.	52
4.3 P@1 Score Regarding to the Number of Negative Candidates Per Question Used in the Training. Each Model Is Initialized with the STS Model.	60
5.1 Two Examples from OkVQA: the Middle Column Are Predictions by Two Baselines and One by Our Proposed Visual-Retriever-Reader Pipeline. The Left Column Is Relevant Knowledge and the Corresponding Captioning of Images.	69
5.2 The Overall Process of Knowledge Corpus Creation. The Question First Combines the Answers One by One to Form a Query, and then the Query Is Sent to the Google Search API to Retrieve the Top 10 Webpages. The Knowledge Is Obtained from the Snippet with Further Processing. Finally, We Integrate the Knowledge Into the Corpus. As Shown on the Search Result Page, the Black Boxes Represent Webpages, and the Red Boxes Represent Snippets.	70

Figure	Page
5.3 Comparison Between Standard DPR, Image-DPR, and Caption-DPR: while the Context Encoder Is the Same for Three Models, in Standard BERT(Left), the Question Encoder Only Takes a Question as Input, the Image-DPR(Middle) Takes Both Question and Image as Input, the Caption-DPR (Right) Takes the Question and the Caption as Input....	73
6.1 The Image Shows the Empire State Building, and the Question Asks if It Is the Tallest Building in “the City” (New York). K1 Is Retrieved by Using Only the Image, K2 Is Retrieved by Only Using the Question, and K3 Is Retrieved Using Both Image and the Question. Only K3 Can Be Used to Answer the Question Correctly.....	78
6.2 Dataset Creation Procedure for ReMuQ. We Use WebQA as Source of the Raw Data. The Multimodal-Query in ReMuQ Is the Combination of an Image and the Question from WebQA where the Overlapped Information with the Image Is Removed. The Ground Truth Knowledge of ReMuQ Is the Answer from WebQA. The Corpus Consists of All Answers and the Distracted Knowledge Candidates Given in ReMuQ... .	83
6.3 Overall Architecture of Our Proposed ReViz. Our Model Consists of a Vision-Language Transformer that Encodes the Image and Text, Mean- while the Knowledge Encoder Projects the Knowledge Into Knowledge Embedding. During Inference, Our Model Selects the Knowledge from the Corpus that Has the Largest Relevance Score with the Image-Text Embedding.	84

Figure	Page
6.4 Figure on the Left Shows an Example of the WIT Dataset, Crawled from Wikipedia. Figure on the Right Shows Our Constructed (T, I, K) Triplet: T Is a Sentence from the Passage and the Words Overlapped with the Title/caption Is Masked; K Is the Remaining Passage after Removing the Sentence.	86
6.5 Evaluation of Out-Of-Domain Performances of ReViz and ReViz+VL-ICT. For OKVQA, We Retrieve Knowledge from GS-112K Corpus. VL-ICT Substantially Improves the Generalization of ReViz. $X \rightarrow Y$ Denotes Using X as the Training Domain and Y as the Testing Domain.	90
6.6 Effect of the Masking Ratio of Sentences in VL-ICT Task on ReViz’s Performance on OKVQA Task. We Use GS112K as the Knowledge Corpus.	93
6.7 Comparison of Captioning-Dependent Retrievers Using Generated Captions and Ground Truth Captions. The Ground Truth Captions Always Lead to Better Performance Than Generated Caption.	94
7.1 Comparison Among Generative and Extractive Readers on Different Length of the Question and Context. Left Part for IID and Right Part for OOD Datasets. Dash Line for Extractive and Solid Line for Generative Readers.	102
9.1 An Example from the HotpotQA Dataset, where the Question Should Be Answered by Combining Supporting Facts(SP) from Two Passages. In the SP, the First String Refers to the Title of Passage, and the Second Integer Means the Index of the Sentence.	116

Figure	Page
9.2 The Architecture of Two-In-One Model for Passage Ranking and Relevant Sentence Selection. For HotpotQA Dataset, K Is Two.	118
10.1 Three Types of Classification Models: Single Classification: All Answer Choices Are Concatenated in One Single Input, and the Model Classifies One Label from Answer Choices. Multiple Choice Question Answering: Each Answer Is Independently Given to the Model and the Model Predicts a Score for Each Answer. For the Answer Extraction Model, It Classifies the Start and End Tokens from the Answer Choices (the Classification Layer of the End Token Is Not Shown in the Figure).	129
12.1 (top) The workflow for generating Alternative Answer Set (AAS) for VQA datasets (bottom) An example from GQA dataset showing semantically valid AAS for the answer ‘batter’ generated using above workflow	142
12.2 Union AAS Score of Different Value of K	145

Chapter 1

INTRODUCTION

1.1 Overview for Information Retrieval and Question Answering

The expansion and accessibility of the internet have not only accelerated the spread and generation of information and knowledge but also intensified people's thirst for knowledge and frequency of information acquisition. Additionally, the growing utilization of multimodal data, such as images, videos, and audio, has broadened the scope of information beyond text. Consequently, information retrieval (IR) and question answering (QA) have become vital tasks in daily life, incorporating multimodal data. IR aims to identify relevant information within a vast corpus containing both textual and multimodal content to address query-based needs, while QA endeavors to comprehend questions and provide accurate, concise responses, often requiring the processing and interpretation of multimodal data. This thesis seeks to advance the development of IR and QA systems and integrate them to address real-world challenges.

The most common task and widely-used application that involve both IR and QA is open domain question answering (ODQA) Chen *et al.* (2017a). Unlike traditional QA Rajpurkar *et al.* (2016), where both the question and the passage containing the answer are provided, ODQA is more close to real-life situations since people often ask open-ended questions without providing the relevant passage, necessitating the retrieval of pertinent information from external sources before providing an accurate answer. Due to its broad real-life applications, ODQA has garnered increasing interest and attention, with numerous benchmarks proposed (Cohen *et al.*, 2018; Khot *et al.*,

2020; Ahmad *et al.*, 2019; Guo *et al.*, 2021). Nowadays, large language models (LLMs) based on transformer architecture and self-supervised learning, such as next word prediction, are pretrained on vast amounts of data. This enables LLMs to serve as knowledge bases, allowing them to answer open-ended questions without retrieving relevant information. Despite their impressive performance, the two-step approach of retrieve-and-then-read still outperforms the pure LLMs and is also considered more reliable and interpretable.

Information retrieval (IR) has a long history, with the first automated IR systems dating back to the 1950s. In this thesis, we refer to IR methods or systems as “retrievers”. Traditional retrievers primarily rely on term-matching, searching for information that overlaps most with the query terms. Algorithms like TF-IDF and BM25 (Robertson and Zaragoza, 2009a) are strong and efficient in this category, taking into account the importance and frequency of terms in queries and documents. However, they suffer from term-mismatch issues and lack semantic understanding of queries and documents Chang *et al.* (2020). Using neural models to represent the concatenation of queries and passages offers a promising solution for semantic matching (Nogueira and Cho, 2019; Banerjee and Baral, 2020). Although these methods are effective for small-scale ranking, they are not applicable to large-scale retrieval. Recently, dual-encoder architecture retrievers based on language models (LMs), such as BERT (Devlin *et al.*, 2019a), have demonstrated their ability to perform semantic matching on a larger scale (Karpukhin *et al.*, 2020b; Guu *et al.*, 2020b; Lewis *et al.*, 2020a). These neural retrievers (NRs) use two LMs to compute vector representations of queries and documents. They are trained to maximize the dot product between the two representations for documents that best answer a query. Despite the success of neural retrievers, they still face numerous challenges (Thakur *et al.*, 2021b; Sciavolino *et al.*, 2021).

In addition, query can be asked beyond text and similar for the context, they can be multi-modality. Multimodal query has special advantage over single modality since in some cases, single modality (i.e. format) such as text may not be adequate for a query to convey all relevant cues. For instance if someone spots a flower and want to find where to buy it, this person would need to know the name of the flower – without knowing the name of the flower, it would be difficult to use a text query to find relevant information. However, multi-modal queries that combine an image of the flower taken with a mobile camera and the text phrase “shops that sell this flower” allow a user to convey such information. In the last couple of years, highly impactful technological advances have been made in the field of multi-modal representation learning, and using these learned representations have been used to improve zero-shot image classification (CLIP (Radford *et al.*, 2021a)), text-to-image synthesis (DALL-E (Ramesh *et al.*, 2021)), image captioning (OSCAR (Li *et al.*, 2020b)), and prompt-based vision-and-language tasks (FLAMINGO (Alayrac *et al.*, 2022)). These advances in shared image-text representations are also being used for information retrieval.

This thesis also investigates question answering (QA), a crucial task that serves as a benchmark for assessing the reading comprehension abilities of intelligent systems, with direct applications in search engines (Kwiatkowski *et al.*, 2019a) and dialogue systems (Reddy *et al.*, 2019; Choi *et al.*, 2018). Advanced QA models are primarily built upon pretrained language models, and when fine-tuned on downstream tasks, they achieve state-of-the-art performance across various tasks. These models can be classified into two main types. Extractive readers (Seo *et al.*, 2017; Devlin *et al.*, 2019a) are commonly used for QA tasks and identify the start and end positions of the answer within the context. Generative readers (Raffel *et al.*, 2020; Lewis *et al.*, 2020b; Izacard and Grave, 2021), on the other hand, generate answers by autoregressively

predicting tokens and have also demonstrated exceptional performance. Along with the fine-tuning, large language models such as GPT-3, PaLM, and the latest model ChatGPT exhibit impressive performance on diverse QA tasks using a zero-shot paradigm. Their performance can be further enhanced through in-context learning, which involves providing instructions and a few examples. Despite their remarkable performance, concerns still persist regarding issues like hallucination and limited reasoning capabilities.

In the following section, we present the thesis outline, organized by individual chapters that detail the limitations of previous systems or methods, as well as our contributions to addressing these shortcomings.

1.2 Thesis Outline and Contribution

- In Chapter 2, we show the limitations of existing information retrieval systems when confronted with the unique challenges posed by biomedical domains. To tackle this issue, we introduce a novel neural retriever called Poly-DPR, a multi-vector dense passage representation model designed to retrieve both short and long biomedical articles effectively. We also incorporate two pretraining tasks that leverage large-scale, freely available data to boost Poly-DPR’s performance. Furthermore, we develop a Template-based Question Generation method to produce domain-specific questions at scale, thereby enhancing Poly-DPR’s proficiency in handling biomedical queries. Our extensive experiments illustrate that Poly-DPR surpasses the performance of prior information retrieval systems in this context.
- In Chapter 3, we highlight the importance of efficiency, generalization, and robustness in information retrieval systems. To achieve greater efficiency, we

propose a joint training method that reduces the indexing memory of neural retrievers. To enhance generalization and robustness, we combine our efficient retriever with a sparse retriever. Additionally, we utilize adversarial attack techniques to create testbeds for evaluating the robustness of our information retrieval system. We believe that this benchmark is a valuable resource for future research on robust information retrieval systems.

- In Chapter 4, we study neural rerankers, models designed to rank a relatively small set of candidates rather than all candidates in the entire knowledge corpus. We observe that the existing training objectives for neural rerankers treat all negative candidates equally, which does not fully exploit the potential of these candidates. Consequently, we introduce the concept of training a neural reranker by distinguishing negative candidates and propose a scoring approach to assign scores to these candidates. These scores serve as signals to train the neural reranker. By comparing our method with standard training methods, we demonstrate the importance and effectiveness of differentiating negative candidates when training neural rerankers.
- In Chapter 5, we study multimodal query information retrieval tasks (MQIR), i.e. queries containing information split across image and text inputs, a challenging task that differs from previous work on cross-modal retrieval. We introduce the first neural information retrieval systems in the field of knowledge-based visual question answering. We propose two types of neural retrievers and demonstrate state-of-the-art performance on the OkVQA benchmark.
- In Chapter 6, we point out that there lacks of an annotated datasets for MQIR task, thus we curate a new dataset called ReMuQ for benchmarking progress on this task. ReMuQ requires a system to retrieve knowledge from a large

corpus by integrating contents from both text and image queries. We introduce a retriever model “ReViz” that can directly process input text and images to retrieve relevant knowledge in an end-to-end fashion without being dependent on intermediate modules such as object detectors or caption generators. We introduce a new pretraining task that is effective for learning knowledge retrieval with multimodal queries and also improves performance on downstream tasks. We demonstrate superior performance in retrieval for on two datasets (ReMuQ and OK-VQA) under zero-shot settings as well as further improvements when finetuned on these datasets.

- In Chapter 7, we conduct a thorough analysis of two commonly-used reader architectures: extractive and generative readers. Based on our analysis, we provide a set of guidelines for selecting the appropriate reader architecture for a given domain. Specifically, we find that generative models are well-suited for answering questions that require long-context comprehension, while extractive models are more effective for generalization and identifying rare answers.
- In Chapter 8, we comprehensively evaluate different data modification strategies and their impact on in-domain and out-of-domain performance as well as adversarial robustness. We also provide insights on the relationship between generalization and adversarial defense. Our findings suggest that additional data improves both out-of-domain accuracy and adversarial robustness, but data filtering can hurt out-of-domain accuracy in certain tasks.
- In Chapter 9, we focus on two important sub-tasks of open-domain question answering: selecting relevant passages and sentences. Unlike complex existing frameworks which use separate models for each task, we propose a simple yet effective framework to jointly rank passages and select sentences, with consis-

tency and similarity constraints to promote interaction between the tasks. Our framework achieves competitive results on the HotpotQA dataset, outperforming the baseline by 28% in exact matching of relevant sentences.

- In Chapter 10, we introduce LogiGLUE, a multi-task logical reasoning benchmark with 8 datasets covering various logical reasoning types, including multiple choice question answering, natural language inference, and fact verification. We fine-tune RoBERTa and T5 on LogiGLUE with different task formalizations and observe that generative models (T5) are not as good as the classification model (RoBERTa). To improve the logical reasoning capacity of generative model, we train a multi-tasking model on LogiGLUE, and results in a model named LogiT5 which show improved performance than the vanilla T5 model.
- In Chapter 11, we present the first extractive model for visual question answering that can leverage retrieved knowledge, resulting in better generalization compared to previous models that rely on the training set answers.
- In Chapter 12, we highlight the drawback of current evaluation metrics for VQA tasks, which do not consider semantic similarity and may unfairly penalize VQA models that provide answers that are semantically close to the ground truth. To overcome this limitation, we introduce Alternative Answer Sets (AAS) of ground-truth answers, generated automatically using NLP tools. We also propose a semantic metric based on AAS and evaluate various models on multiple datasets. Our results show that the AAS metric outperforms existing metrics, and our human study aligns well with the AAS scores.

Chapter 2

TEXT RETRIEVER: IMPROVING BIOMEDICAL INFORMATION RETRIEVAL WITH NEURAL RETRIEVERS

Information retrieval (IR) is widely used in commercial search engines and is crucial for tasks like open-domain question answering and open-domain dialogue tasks that demand external knowledge. Furthermore, IR can help tackle complex problems in natural language processing (NLP) by obtaining pertinent information, thereby enhancing performance and interpretability. The biomedical domain has seen increased reliance on IR due to the exponential growth of electronic information availability (Shortliffe *et al.*, 2014). Conventional biomedical IR has depended on term-matching algorithms such as TF-IDF and BM25 (Robertson and Zaragoza, 2009a), which look for documents containing terms specified in the query. For instance, in Table 2.1, the first example illustrates BM25 retrieving a sentence with the word “Soluvia” from the question. However, term-matching faces challenges, especially when dealing with terms that have varying meanings in different contexts (as observed in the second example) or when essential semantics from the question are not factored into the retrieval process (as seen in the third example, where the term “how large” is not represented in the answer retrieved by BM25).

Since these failure modes can have a direct impact on downstream NLP tasks such as open-domain question answering (ODQA), there has been interest in developing neural retrievers (NR) (Karpukhin *et al.*, 2020b). NRs which represent query and context as vectors and utilize similarity scores for retrieval, have led to state-of-the-art performance on ODQA benchmarks such as Natural Questions (Kwiatkowski *et al.*, 2019c) and TriviaQA (Joshi *et al.*, 2017). Unfortunately, these improvements on

standard NLP datasets are not observed in the biomedical domain with NRs.

Recent work provides useful insights to understand a few shortcomings of NRs. Thakur *et al.* (2021b) find NRs to be lacking at exact word matching, which affects performance in datasets such as BioASQ (Tsatsaronis *et al.*, 2015) where exact matches are highly correlated with the correct answer. Lewis *et al.* (2021b) find that in the Natural Questions dataset, answers for 63.6% of the test data overlap with the training data and DPR performs much worse on the non-overlapped set than the test-train overlapped set. In this work, we found this overlap to be only 2% in the BioASQ dataset, which could be a potential reason for lower performance of NR methods. We also discovered that NRs produce better representations for short contexts that for long contexts – when the long context is broken down into multiple shorter contexts, performance of NR models improves significantly.

In Luo *et al.* (2022c), we seek to address these issues and improve the performance of neural retrieval beyond traditional methods for biomedical IR. While existing systems have made advances by improving neural re-ranking of retrieved candidates (Almeida and Matos, 2020; Pappas *et al.*, 2020), our focus is solely on the retrieval step, and therefore we compare our neural retriever with other retrieval methods. Our method makes contributions to three aspects of the retrieval pipeline – question generation, pre-training, and model architecture.

Our first contribution is the “**Poly-DPR**” model architecture for neural retrieval. Poly-DPR builds upon two recent developments: Poly-Encoder Humeau *et al.* (2020) and Dense Passage Retriever (Karpukhin *et al.*, 2020b). In DPR, a question and a candidate context are encoded by two models separately into a contextual vector for each, and a score for each context can be computed using vector similarity. On the other hand, Poly-Encoder represents the query by K vectors and produces context-specific vectors for each query. Instead, our approach Poly-DPR represents each

Question	Answer	BM25	DPR
What is Soluvia?	Soluvia by Becton Dickinson is a microinjection system for intradermal delivery of vaccines.	The US FDA approved Sanofi Pasteur's FluZone Intradermal influenza vaccine that uses a new microinjection system for intradermal delivery of vaccines (Soluvia, Becton Dickinson).	Internet-ordered viagra (sildenafil citrate) is rarely genuine.
Is BNN20 involved in Parkinson’s disease?	BNN-20 could be proposed for treatment of PD	Rare causes of dystonia parkinsonism.	BNN-20 could be proposed for treatment of PD
How large is a lncRNAs?	lncRNAs are defined as RNA transcripts longer than 200 nucleotides that are not transcribed into proteins.	lncRNAs are closely related with the occurrence and development of some diseases.	An increasing number of long noncoding RNAs (lncRNAs) have been identified recently.

Table 2.1: Illustrative examples from the BioASQ challenge along with the context retrieved by two methods BM25 and DPR.

context by K vectors and produces *query-specific vectors* for each context. We further design a simple inference method that allows us to employ MIPS (Shrivastava and Li, 2014) during inference.

Next, we develop “**Temp-QG**”, a template-based question generation method which helps us in generating a large number of domain-relevant questions to mitigate the train-test overlap issue. TempQG involves extraction of templates from in-domain questions, and using a sequence-to-sequence model Sutskever *et al.* (2014) to generate

questions conditioned on this template and a text passage.

Finally, we design two new pre-training strategies: “**ETM**” and “**RSM**” that leverage our generated dataset to pre-train Poly-DPR. These tasks are designed to mimic domain-specific aspects of IR for biomedical documents which contain titles and abstracts, as opposed to passage retrieval from web pages Chang *et al.* (2020). Our pre-training tasks are designed to be used for long contexts and short contexts. In both tasks, we utilize keywords in either query or context, such that the capacity of neural retrievers to match important keywords can be improved during training.

Armed with these three modules, we conduct a comprehensive study of document retrieval for biomedical texts in the BioASQ challenge. Our analysis demonstrates the efficacy of each component of our approach. Poly-DPR outperforms BM25 and previous neural retrievers for the BioASQ challenge, in the small-corpus setting. A hybrid method, which is a simple combination of BM25 and NR predictions, leads to further improvements. We perform a post-hoc error analysis to understand the failures of BM25 and our Poly-DPR model. Our experiments and analysis reveal aspects of biomedical information retrieval that are not shared by generic open-domain retrieval tasks. Findings and insights from this work could benefit future improvements in both term-based as well as neural-network based retrieval methods.

2.1 Overview of Existing Retriever

In general, the modern text retrievers can be categorized in five classes (adapted from (Thakur *et al.*, 2021b)). **Lexical retrievers** such as BM25 Robertson and Zaragoza (2009a) are based on token-matching between two high-dimensional sparse vectors. The sparse vectors are represented based on the frequency of the terms in documents and thus does not require any annotated training data. Regardless of the simplicity of the algorithms, such methods perform well on new domains (Thakur *et al.*,

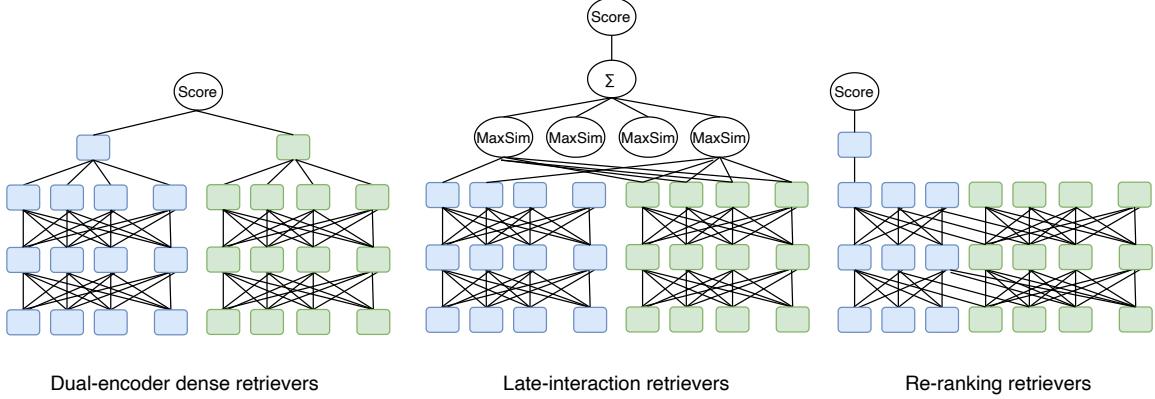


Figure 2.1: Architectures of three major types of retrievers. For simplicity, some lines in the figures are not drawn. Blue blocks represent the encoding for question, and the green blocks represent context or documents.

2021b). **Dual-encoder dense retrievers** consists of two encoders where the query encoder and context encoder generate a single dense vector representation for query and context respectively. Then the score can be computed by inner-dot product or cosine-similarity between the two representations (Karpukhin *et al.*, 2020b; Xiong *et al.*, 2020; Hofstätter *et al.*, 2021). Language models such as BERT Devlin *et al.* (2019a) are preferred choices for encoders. **Sparse retrievers** use sparse representations instead of dense representations for query and document (Dai and Callan, 2020; Zhao *et al.*, 2021; Nogueira *et al.*, 2019). **Late-interaction retrievers** different from dense retrievers who use sequence-level representations of query and document, they use token-level representations for the query and passage: a bag of multiple contextualized token embeddings Khattab and Zaharia (2020). The late-interactions are aggregated with sum of the max-pooling query term and a dot-product across all passage terms. **Re-ranking retrievers** include two stages, coarse-search by efficient methods (e.g. BM25) and fine-search by cross-attentional re-ranking models. The re-ranking model takes input as the concatenation of the query and one candidate given by the first stage and produce a score based on the cross representation (e.g. the [CLS] token), and such process is repeated for every candidate, and finally re-rank candidates based

on the generated scores.

Without changing the architectures, different efforts have been made toward learning better representation of dense vectors and improving the efficiency in terms of training resources as well as short inference time. One way to improve the representation of dense vectors is to construct proper negative instances to train a neural retriever. In-batch negative training is a frequently used strategy to train dense retrievers, and the larger the batch size is, the better performance a dense retriever can achieve (Karpukhin *et al.*, 2020b; Qu *et al.*, 2021b). Using hard negative candidates is better than using random or simple in-batch negative samples, for example, Karpukhin *et al.* (2020b) mine negative candidates by BM25 and (Xiong *et al.*, 2020) mine negative candidates from the entire corpus using an optimized dense retriever. Hofstätter *et al.* (2021) selects the negative candidates from the same topic cluster, such a balanced topic aware sampling method allows the training with small batch size and still achieves high quality dense representation. ColBert Khattab and Zaharia (2020) is proposed to improve the efficiency of the ranking model. Since every token can be pre-indexed, it prevents inference time from getting representation of context. While Colbert is faster than single-model, it is slower compared to dual-models, thus, it is not suitable for retrieval at large scale. On the other hand, Nogueira *et al.* (2019) shortens the inference time by using sparse representation for queries.

Next, we give a detailed description of some fundamental retriever which inspire the proposed neural retriever introduced in §2.2.

2.1.1 Bag-of-Words Retriever: BM25

BM25 Robertson *et al.* (2009), a ranking function that scores the query and document based on the term frequency. The following equation is the one of the most prominent instantiations of the function,

$$score(D, Q) = \sum_{i=1}^n IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgdl})} \quad (2.1)$$

where $IDF(q_i)$ is the inverse document frequency of query term q_i , $f(q_i, D)$ is the frequency of q_i in document D , $|D|$ is the length of the document D , and $avgdl$ is the average length of all documents in the corpus. In practice, $k_1 \in [1.2, 2.0]$ and $b = 0.75$. BM25 is an unsupervised method that generalizes well in different domains Thakur *et al.* (2021b) and has been widely used in search engines.

2.1.2 Neural Retriever: Dense Passage Representation

Dense Passage Representation (DPR) (Karpukhin *et al.*, 2020b) is a neural retriever model belonging to the dual-model family. DPR encodes the query q and the context c into dense vector representations:

$$v_q = E_q(q) [\text{CLS}], \quad v_c = E_c(c) [\text{CLS}]. \quad (2.2)$$

where E_q and E_c are BERT (Devlin *et al.*, 2019b) models which output a list of dense vectors (h_1, \dots, h_n) for each token of the input, and the final representation is the vector representation of special token $[\text{CLS}]$. E_q and E_c are initialized identically and are updated independently while being trained with the objective of minimizing the negative log likelihood of a positive (relevant) context. A similarity score between q and each context c is calculated as the inner product between their vector representations:

$$\text{sim}(q, c) = v_q^T v_c. \quad (2.3)$$

2.1.3 Neural Retriever: Poly-Encoder

Poly-Encoder (Humeau *et al.*, 2020) also uses two encoders to encode query and context, but the query is represented by K vectors instead of a single vector as in DPR. Poly-Encoder assumes that the query is much longer than the context, which

is in contrast to information retrieval and open-domain QA tasks in the biomedical domain, where contexts are long documents and queries are short and specific.

2.2 Poly-DPR: Multi Vectors Dense Passage Representation

We integrate Poly-Encoder and DPR to use K vectors to represent context rather than query. In particular, the context encoder includes K global features (m_1, m_2, \dots, m_k) , which are used to extract representation $v_c^i, \forall i \in \{1 \dots k\}$ by attending over all context tokens vectors.

$$v_c^i = \sum_n w_n^{m_i} h_n, \text{ where} \quad (2.4)$$

$$(w_1^{m_i}, \dots, w_n^{m_i}) = \text{softmax}(m_i^T \cdot h_1, \dots, m_i^T \cdot h_n). \quad (2.5)$$

After extracting K representations, a query-specific context representation $v_{c,q}$ is computed by using the attention mechanism:

$$v_{c,q} = \sum_k w_k v_c^k, \text{ where} \quad (2.6)$$

$$(w_1, \dots, w_k) = \text{softmax}(v_q^T \cdot v_c^1, \dots, v_q^T \cdot v_c^k). \quad (2.7)$$

Although we can pre-compute K representations for each context in the corpus, during inference, a ranking of the context needs to be computed after obtaining all query-specific context representations. As such, we can not directly use efficient algorithms such as MIPS Shrivastava and Li (2014). To address this challenge, we use an alternative similarity function for inference – the score $\text{sim}_{\text{infer}}$ is computed by obtaining K similarity scores for the query and each of the K representations, and take the maximum as the similarity score between context and query:

$$\text{sim}_{\text{infer}}(q, c) = \max(v_q^T \cdot v_c^1, \dots, v_q^T \cdot v_c^k). \quad (2.8)$$

Using this similarity score, we can take advantage of MIPS to find the most relevant context to a query.

In sum, Poly-DPR differs from Poly-Encoder in two major aspects: (1) K pre-computed representations of context as opposed to K representations computed during inference, and (2) a faster similarity computation during inference.

Hybrid Model We also explore a hybrid model that combines the traditional approach of BM25 and neural retrievers. We first retrieve the top-100 candidate articles using BM25 and a neural retriever (Poly-DPR) separately. The scores produced by these two methods for each candidate are denoted by S_{BM25} and S_{NR} respectively and normalized to the $[0, 1]$ range to obtain S'_{BM25} and S'_{NR} . If a candidate article is not retrieved by a particular method, then its score for that method is 0. For each article, we get a new score:

$$S_{\text{hybrid}} = S'_{\text{BM25}} + S'_{\text{NR}}. \quad (2.9)$$

Finally, we re-rank candidates based on S_{hybrid} and pick the top candidates – for BioASQ performance is evaluated on the top-10 retrieved candidates.

2.3 Pretraining Tasks For Neural Retriever

Masked language modeling (MLM) and next-sentence prediction introduced in BERT (Devlin *et al.*, 2019b) have led to a paradigm shift in the training of neural network models for multiple NLP tasks. significant progress in multiple NLP tasks. MLM operates by masking out some tokens in text and learning representations by training models to predict these masked tokens. “Next sentence prediction” is also employed as a pre-training task, where the model is trained to predict if an input sentence is semantically followed by another input sentence. For text retrieval, pre-training tasks that are more aligned with the retrieval task have been developed. Chang *et al.* (2020) propose Body First Selection (BFS), and Wiki Link Prediction (WLP) for document retrieval. Lee *et al.* (2019a) propose an Inverse Cloze Task

(ICT) task in which a random sentence drawn from a passage acts as a query and the remaining passage as a relevant answer. Guu *et al.* (2020c) show that ICT effectively avoids the cold-start problem. Gao and Callan (2022) uses Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval.

Our aim is to design pre-training tasks specifically for the biomedical domain since documents in this domain bear the (*title*, *abstract*, *main text*) structure of scientific literature. This structure is not commonly found in documents such as news articles, novels, and text-books. Domain-specific pre-training tasks have been designed by Chang *et al.* (2020) for Wikipedia documents which contains hyperlinks to other Wikipedia documents. However most biomedical documents do not contain such hyperlinks, and as such, pre-training strategies recommended by Chang *et al.* (2020) are incompatible with structure of biomedical documents. Therefore, we propose Expanded Title Mapping (ETM) and Reduced Sentence Mapping (RSM), designed specifically for biomedical IR, to mimic the functionality required for open-domain question answering. An overview is shown in Figure 2.3. The proposed tasks work for both short as well as long contexts. In biomedical documents, each document has a title (T) and an abstract (A). We pre-train our models on ETM or RSM and then finetune them for retrieval.

2.3.1 Expanded Title Mapping Pretraining Task.

For Expanded Title Mapping (ETM), the model is trained to retrieve an abstract, given an extended title T' as a query. T' is obtained by extracting top- m keywords from the abstract based on the TF-IDF score, denoted as $K = \{k_1, k_2, \dots, k_m\}$, and concatenating them with the title as: $T' = \{T, k_1, k_2, \dots, k_m\}$. The intuition behind ETM is to train the model to match the main topic of a document (keywords and title) with the entire abstract.

2.3.2 Reduced Sentence Mapping Pretraining Task.

Reduced Sentence Mapping (RSM) is designed to train the model to map a sentence from an abstract with the extended title T' . For a sentence S from the abstract, we first get the weight of each word $W = \{w_1, w_2, \dots, w_n\}$ by the normalization of TF-IDF scores of each word. We then reduce S to S' by selecting the words with the top- m corresponding weights. The intuition behind a reduced sentence is to simulate a real query which usually is shorter than a sentence in a PubMed abstract. Furthermore, S' includes important words based on the TF-IDF score, which is similar to a question including keywords.

2.4 Question Generation For Large Scale Training

Question generation methods have become sophisticated due to the advances in sequence-to-sequence modeling (Sutskever *et al.*, 2014); QG is considered an auxiliary pre-training task for question answering models (Alberti *et al.*, 2019). One set of QG methods can be categorized as ‘Answer-Aware’ QG (Du and Cardie, 2018; Zhao *et al.*, 2018; Dong *et al.*, 2019), in which an answer extraction model first produces potential answers, followed by a question generator which generates a question given the context and a potential answer. Alberti *et al.* (2019) utilizes cycle consistency to verify whether a question-answering model predicts the same answer to the generated question. A second set of QG methods generate questions without conditioning the generator using the answer – for instance, Lopez *et al.* (2020) propose end-to-end question generation based on the GPT-2 model, while Lewis *et al.* (2019); Fabbri *et al.* (2020); Banerjee *et al.* (2021a) generate questions using linguistic and semantic templates. Question paraphrasing Hosking and Lapata (2021) is a related approach for creating augmented training samples. Question generation has also been explored

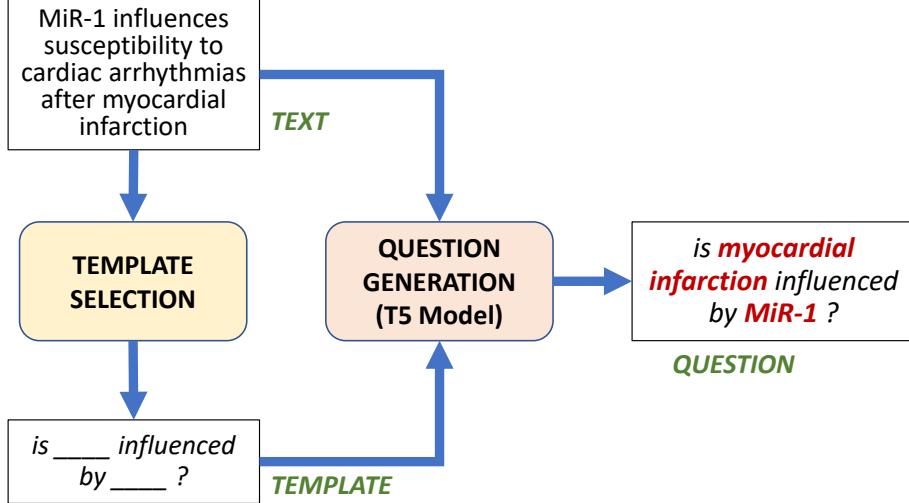


Figure 2.2: Overview of Template-Based Question Generation.

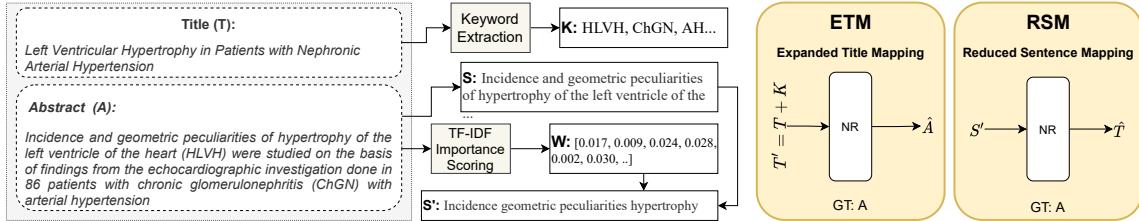


Figure 2.3: Poly-DPR is pre-trained on two novel tasks designed specifically for information retrieval applications. This figure illustrates the sample generation pipeline using the title and abstract from each sample in BioASQ.

in visual question answering, with end-to-end methods Li *et al.* (2018b); Krishna *et al.* (2019) and template-based methods Banerjee *et al.* (2021b). While our proposed question generation method is also template-based, instead of using a pre-defined list of templates designed by humans, our template extraction process is automated.

2.4.1 Template Based Question Generation: TempQG

We propose a template-based question generation approach – *TempQG*, that captures the style of the questions in the target domain. Our method consists of three modules: template extraction, template selection, and question generation.

Template Extraction aims to extract unique templates from which the questions in the training set can be generated. We first use bio-entity taggers from Spacy (<https://spacy.io/>) to obtain a set of entities from the question. We replace non-verb entities having a document frequency less than k with an underscore (_) – this prevents common entities such as “disease”, “gene” from being replaced. For e.g., given the question “*Borden classification is used for which disease?*”, the entity tagger returns [“*Borden classification*”, “*disease*”], but only the first entity clears our frequency-based criteria. As a result, the generated template is “*_ is used for which disease?*”. This process gives us a preliminary list of templates. We then use a question similarity model (which returns a score between $[0, 1]$) to compute the pairwise score between all templates. Templates are assigned to a cluster if they have a minimum similarity of 0.75 with existing templates of a cluster. Once clusters have been formed, we choose either the sentence with the smallest or second-smallest length as the representative template. These representative templates are used for question generation.

Template Selection. Given a text passage, we create a text-template dataset and train the PolyDPR architecture to retrieve a relevant template. After the model is trained, we feed new text inputs to the model, obtain query encoding and compute the inner product with each template. Templates with maximum inner product are selected to be used for QG.

Question Generation (QG). We use a T5 (Raffel *et al.*, 2020) model for generating questions, by using text and template as conditional inputs. To distinguish between these two inputs, we prepend each with the word “template” or “context”, resulting in an input of the form: {“*template*” : *template*, “*context*” : *text*}. Figure 2.2 shows an illustrative example for the template-based question generation method abbreviated

#	Context	Template Generated	Question
1	The lysosomal-membrane protein type 2A (LAMP-2A) which reacts as the receptor for the substrates of chaperone-mediated autophagy (CMA), which should undergo unfolding before crossing the lysosomal membrane and reaching the lumen for degradation.	which receptor is targeted by LAMP-2A?	Which receptor is targeted by LAMP-2A?
2	Is Tokuhashi score suitable for evaluation of life expectancy before surgery in Iranian patients with spinal metastases? One of the most important selection criteria for spinal metastases surgery is life expectancy and the most important system for this prediction has been proposed by Tokuhashi.	what is evaluated with _	What is the Tokuhashi score?
3	Lambert-Eaton myasthenic syndrome (LEMS) is a pre-synaptic disorder of the neuromuscular and autonomic transmission mediated by antibodies to voltage-gated calcium channels at the motor nerve terminal.	_ is diagnosed in which neuromuscular and autonomic pathways?	Lambert-Eaton myasthenic syndrome is diagnosed in which neuromuscular and autonomic pathways?

Table 2.2: Illustrative examples for templates and questions generated by TempQG

as *TempQG*. The context used for generating the questions are any two consecutive sentences in the abstract. Given such a context, we first select 10 unique templates and concatenate each template with the context independently. These are used by the question generation model to produce 10 initial questions; duplicate questions are filtered out.

Generation Quality Analysis Table 2.2 shows examples of selected templates and generated questions. Our template-based generation approach can produce diverse and

domain-style questions using three strategies. **Fill in the blank**: the generator fills the blank in the template by key entities mentioned in the context without changing the template, as shown by Example 1. **Changing partially**: the generator produces questions by using part of the template and ignores some irrelevant part as shown by Example 2. **Ignoring entirely**: the generator ignores the template entirely and generates questions that are not relevant to the given context as shown by Example 3.

2.5 Experiments and Results

Size of Corpus. PubMed is a large corpus containing 19 million articles, each with a title and an abstract. Due to this large corpus size, indexing the entire corpus takes a significantly long time. To conduct comprehensive experiments and to efficiently evaluate the impacts of each proposed method, we construct a small corpus with 133,084 articles in total: 33,084 articles belonging to the training and test sets of BioASQ8, and an additional $100K$ articles that are randomly sampled from the entire corpus.

Length of Context. We use two context lengths for training neural retrievers and indexing the corpus: 128 (short) and 256 (long). We use RSM as the pre-training task for short contexts and either ETM or ICT with long contexts.

Training Setup. We use BioBERT (Lee *et al.*, 2020) as the initial model for both query and context encoders in all experiments. For BM25, we use an implementation from Pyserini (Lin *et al.*, 2021) with default hyperparameters $k=0.9$ and $b=0.4$. We also try $k=1.2$ and $b=0.75$ as used by Ma *et al.* (2021c) and find the default setting to be slightly better. For Poly-DPR, the number of representations K is set as 6 after a hyper-parameter search.

CL	PT	FT	B1	B2	B3	B4	B5	Avg.
Short (128)	-	B	54.48	50.51	53.8	59.06	48.71	53.31
	-	T	62.92	58.79	62.94	70.30	63.39	63.67
	RSM	B	65.94	57.43	61.89	69.01	58.23	62.50
	RSM	A	56.84	55.79	57.52	58.68	55.15	56.80
	RSM	T	64.71	64.92	64.28	73.11	66.29	66.66
Long (256)	-	B	35.69	32.66	32.26	38.28	30.87	33.95
	-	T	63.95	59.51	62.98	66.71	62.80	63.19
	ICT	B	54.44	47.37	52.61	53.69	44.38	50.50
	ETM	B	56.63	46.63	52.79	56.97	49.61	52.53
	ETM	T	64.57	58.51	64.02	68.44	62.60	63.62
	ETM	A	54.44	49.95	48.42	58.15	52.60	52.71
	ICT+ETM	B	51.33	49.43	49.36	53.19	43.58	49.38
	ICT+ETM	T	64.93	58.49	60.18	69.42	64.87	63.58

Table 2.3: Effect of pre-training tasks (PT) and fine-tuning datasets (B: BioASQ, T: TempQG and A: AnsQG) on the performance of Poly-DPR with two context lengths (CL) on the BioASQ small corpus test set. Bi stands for the i^{th} batch in the testing sets.

2.5.1 Results

Effect of Pre-Training Tasks and Fine-Tuning Datasets. Table 2.3 shows results when Poly-DPR is trained with different methods of pre-training and different fine-tuning datasets. Both RSM and ETM lead to improvements even when the finetuning task has only a limited amount of supervised data, i.e. BioASQ. When compared to Poly-DPR trained without any pre-training, RSM improves by $\sim 9\%$ and ETM by $\sim 18\%$. ETM is better than the existing pre-training method ICT (Lee *et al.*, 2019a) by $\sim 2\%$. When the size of fine-tuning set is large, i.e. with our question generation method (TempQG), the gains due to pretraining are higher with short

contexts than with large contexts. We believe this to be a result of the finetuning dataset in the long-context setting being significantly larger than the pre-training dataset, thereby having a larger effect on the training process¹.

We also see that when Poly-DPR is only trained on BioASQ, the performance with small contexts is much better than with long contexts (53.31% vs 33.95%). This suggests that Poly-DPR trained on the small corpus finds it difficult to produce robust representations for long contexts. On the other hand, the performance of Poly-DPR variants trained on TempQG is close for short and long contexts, which suggests that large-scale relevant training data improves representations.

Comparison with Baselines. Table 2.4 shows a comparison between baselines and our best model (Poly-DPR with short context (128) pre-trained with RSM and finetuned on TempQG). Note that our model is only trained on datasets acquired from the small corpus. However, we evaluate the same model on the large corpus test set.

In the **small corpus setting**, it can be seen that our model outperforms all existing methods in the small corpus setting, and is better than DPR by 13.3% and 20.8% in short (128) and long (256) context lengths respectfully. In the **large corpus setting**, our method is better than GenQ (Ma *et al.*, 2021c) on all five test sets. This shows that our method, which uses 10 million generated samples is better than GenQ which uses 83 million samples for training, thus showing the effectiveness of our template-based question generation method. Although our method performs better than BM25 on B1, B2, B5, the average performance is slightly worse (-1.17%). For the hybrid method, we apply our best Poly-DPR model to index the entire corpus, and use the procedure as described in Sec 2.2. Our hybrid method which combines BM25 and Poly-DPR, is better than all existing methods.

Model		B1	B2	B3	B4	B5	Avg.
<i>Small Corpus</i>							
BM25 Robertson and Zaragoza (2009a)		62.15	61.30	66.62	74.14	61.30	65.10
DPR ₁₂₈ Karpukhin <i>et al.</i> (2020b)		54.48	50.51	53.80	59.06	48.71	53.31
DPR ₂₅₆		44.86	41.18	40.25	47.78	40.42	42.89
P-DPR ₁₂₈ (Ours)		64.71	64.92	64.28	73.11	66.29	66.66
P-DPR ₂₅₆ (Ours)		64.57	58.51	64.02	68.44	62.60	63.62
Hybrid (DPR ₁₂₈)		66.55	61.29	68.08	72.91	60.30	65.83
Hybrid (P-DPR ₁₂₈)		66.30	64.90	69.54	75.71	64.82	68.25
<i>Large Corpus</i>							
BM25		28.50	27.82	37.97	41.91	35.42	34.32
GenQ Ma <i>et al.</i> (2021c)		28.90	20.30	30.70	29.00	33.10	28.40
P-DPR ₁₂₈ (Ours)		35.10	29.07	32.74	33.31	35.54	33.15
Hybrid (P-DPR ₁₂₈)		30.02	31.31	39.79	42.18	37.99	36.26
<i>Large Corpus SOTA (Re-ranking)</i>							
PAKazaryan <i>et al.</i> (2020)		35.91	39.45	52.73	41.15	52.02	44.25
bioinfo-4 Almeida and Matos (2020)		38.23	36.86	51.08	46.77	50.98	44.78
AUEB-4 Pappas <i>et al.</i> (2020)		5.47	7.23	53.29	49.92	49.53	33.09

Table 2.4: Comparison between our Poly-DPR (P-DPR) with baseline methods in the small corpus and large corpus settings. The bottom section shows performance of existing methods that make improvements in the re-ranking method.

We also report state-of-the-art (SOTA) results reported on the BioASQ8 leader-board. These approaches are a combination of retrieval and improved re-ranking methods. Since this paper is concerned with improving retrieval and does not study re-ranking, we do not compare our methods directly with these approaches, but report them for completeness.

Index Unit	Mem.	Time	B1	B2	B3	B4	B5	Avg.
2-sents	21.0 G	321	64.71	64.92	64.28	73.11	66.29	66.66
128-chunk	8.1 G	206	65.16	63.24	63.72	72.13	65.29	65.91
256-chunk	4.5 G	192	63.76	59.71	62.70	67.21	64.17	63.51
Full	2.8 G	101	61.92	57.84	60.01	61.11	62.66	60.71
2-sents	21.0 G	321	64.65	59.21	63.65	70.90	65.97	64.88
128-chunk	8.1 G	206	64.11	58.08	64.15	69.90	63.16	63.88
256-chunk	4.5 G	192	64.57	58.51	64.02	68.44	62.6	63.62
Full	2.8 G	101	60.06	56.38	61.99	65.01	59.63	60.61

Table 2.5: Two best NR models in short and long context: the first block is Poly-DPR pretrained with RSM and fine-tuned on TempQG (short); the second block is Poly-DPR pretrained with ETM and fine-tuned on TempQG (long).

NT	B1	B2	B3	B4	B5	Avg.
1	67.21	62.43	66.49	72.15	61.55	65.96
5	66.76	62.19	66.41	71.55	64.33	66.25
10	64.71	64.92	64.28	73.11	62.29	66.66

Table 2.6: Effect of number of templates (NT) on performance.

We provide ablation studies of different hyper-parameters on model performance. Results are reported on the small corpus.

Granularity of Indexing. Here we examine the impact of indexing units. We conjecture that the representation produced with a shorter indexing unit is better than the one with a longer indexing unit, and thus an NR should perform better if the indexing unit is short. To verify this, we use our best Poly-DPR models that are trained in short and long context settings. We compare four indexing units, **2-sents**: two consecutive sentences, **128 chunk**: a chunk with maximum length of 128 tokens

K	B1	B2	B3	B4	B5	Avg.
0	62.06	61.81	61.85	66.69	61.30	62.74
6	62.92	58.79	62.94	70.30	63.39	63.67
12	65.22	60.86	62.59	70.50	66.21	65.08
0	61.70	58.28	58.62	67.33	61.48	61.48
6	63.95	59.51	62.98	66.71	62.80	63.19
12	63.83	57.81	62.72	70.00	63.64	63.60

Table 2.7: Comparison among different values of K for Poly-DPR in both short and long context settings.

that includes multiple consecutive sentences, **256 chunk**: a chunk with maximum length of 256 tokens that includes multiple consecutive sentences, and **512 chunk**: the entire article including title and abstract, and we use 512 tokens to encode each article. The results are shown in Table 2.5; we see that the smaller indexing units yield better performance, even for the model that is trained in long context setting. We also present the memory (*Mem.*) and inference time (*Time*) which depend upon the choice of indexing unit. The inference time refers to the number of seconds taken to retrieve 10 documents for 100 questions. Table 2.5 shows that a smaller indexing unit requires more memory and longer inference time. Thus, there is a trade-off between retrieval quality and memory as well as inference time. Future work could explore ways to improve the efficiency of neural retrievers to mitigate this trade-off.

Number of Templates for Generating Questions We study three values for the number of templates, 1, 5, and 10, and report the results for Poly-DPR in Table 2.6. We see that training Poly-DPR on questions generated from one template is already better than BM25. While increasing the number of templates yields better performance, the improvement is relatively small, and we conjecture that this could

be due to lower-quality or redundant templates. A question filtering module can be used to control the quality of the questions as shown in previous work (Alberti *et al.*, 2019).

Number of Context Representations Poly-DPR encodes a context into K vector representations. We study the effect of three values of K (0, 6, and 12) on model performance, both with short (128) and long (256) contexts. All models are trained directly on the TempQG without pretraining. Table 2.7 shows that a larger K value yields better performance. This observation is aligned with Humeau *et al.* (2020).

2.5.2 Error Analysis

To better understand the differences between BM25 and NR, we study their failure modes. From the BioASQ test set, we select questions on which either BM25 or Poly-DPR perform poorly, and categorize these failure cases (see Table 2.8).

Failures Cases of BM25. We found 91 failure cases on which the MAP score of BM25 is 0 for 41 cases, and the performance of BM25 is at least 0.5 less than Poly-DPR for 50 cases. Upon manual inspection, we identify three common categories of these failures. **B1:** questions contain keywords with typographical errors. **B2:** questions mention multiple entities related to each other. BM25 may fail to retrieve documents that connect these entities. **B3:** questions mention conceptual properties of entities and answers are values. For example, "how large" is a conceptual property and "200" is the answer value. BM25 retrieves documents related to the entities in questions but not contain the answer.

Failure cases of Poly-DPR. There we 55 failure cases of Poly-DPR, including 23 cases with 0 MAP score and 32 case where the score for BM25 is at least 0.5

	Question	Explanation
B1	What is minodixil approved for?	minodixil is a typo, the correct one is minoxidil
B2	List 5 proteins with antioxidant properties?	BM25 fails to connect proteins and antioxidant properties, and retrieves documents all related to antioxidant, however, they are not about proteins nor antioxidant proteins.
B3	How large is a lncRNAs?	BM25 retrieves document about lncRNAs but not about how large it is.
P1	What is Xanamem?	NR fails to retrieve any document related to Xanamem, rather, it retrieves documents that lexical similar to Xanamem such as Ximenia, Xadago, and Xenopus.
P2	Does an interferon (IFN) signature exist for SLE patients?	NR ranks documents about interferon higher than documents of SLE patients and documents of both. In the retrieved documents, interferon appears rather frequently.

Table 2.8: Examples of the common failure modes of BM25 and Poly-DPR.

better than Poly-DPR. There are two common failure modes of Poly-DPR. **P1:** questions are simple but focused on rare entities which Poly-DPR fails to retrieve. This conforms with the finding that NR performs significantly worse than BM25 on entity-questions (Sciavolino *et al.*, 2021). We find that for such questions, retrieved entities and entities in the question are lexical similar or have overlapping substrings, which in turn could be due to the WordPiece embeddings (Wu *et al.*, 2016) used in BERT. **P2:** Questions mention multiple entities. Articles that contain frequent entities are ranked higher than articles that include *all* entities in the question.

2.6 Discussion and Summary

We show that DPR, a neural retriever, is unable to surpass BM25 on biomedical benchmarks such as BioASQ. We address this drawback of NRs with a three-pronged approach with Poly-DPR: a new model architecture, TempQG: a template-based question generation method, and two new pre-training tasks designed for biomedical documents. TempQG can generate high quality domain-relevant questions which positively impact downstream performance. While in this paper, we apply TempQG to a small corpus of 100,000 PubMed articles, we show that this method can surpass neural retrievers when trained on small or large corpora. Our model achieves better performance than BM25 in the small corpus setting, but it falls short by $\sim 1\%$ in the large corpus setting. However, we show that a hybrid model combining our approach and BM25 is better than all previous baselines on the entire corpus. In the future, applying our question generation methods to the entire PubMed corpus, and combining our approach with improved re-ranking techniques could potentially result in further improvement.

Chapter 3

LIGHT HYBRID RETRIEVER FOR EFFICIENCY AND GENERALIZATION

Information retrieval (IR) strives to locate pertinent documents in response to a query, typically from a vast corpus. Consequently, efficiency is a crucial characteristic of IR systems (Min *et al.*, 2021; Bruch *et al.*, 2022), encompassing both latency and memory usage. An efficient system should identify relevant documents quickly and utilize minimal memory. Additionally, an IR system ought to be versatile, as queries and associated documents may originate from various domains, including science, biomedical, and sports.

The classical IR methods, such as BM25 (Robertson *et al.*, 2009), produce sparse vectors for question and documents based on bag-of-words approaches. Recent research pays attention toward building neural retrievers which learn dense embeddings of the query and document into a semantic space (Karpukhin *et al.*, 2020a; Khattab and Zaharia, 2020). Sparse and dense retrievers have their pros and cons, and the hybrid of sparse and dense retrievers can take advantage of both worlds and achieve better performance than individual sparse and dense retrievers. Therefore, hybrid retrievers are widely used in practice (Ma *et al.*, 2021e; Chen *et al.*, 2021).

Previous hybrid retrievers are composed of indexing-heavy dense retrievers, in this work, we study the question “*Is it possible to reduce the indexing memory of hybrid retrievers without sacrificing performance?*” To answer this question, in Luo *et al.* (2022b), we reduce the memory by using the state-of-the-art indexing-efficient retriever, DrBoost (Lewis *et al.*, 2021a), a boosting retriever with multiple “weak” learners. Compared to DPR (Karpukhin *et al.*, 2020a), a representative dense retriever, DrBoost reduces the indexing memory by 6 times while maintaining the performance.

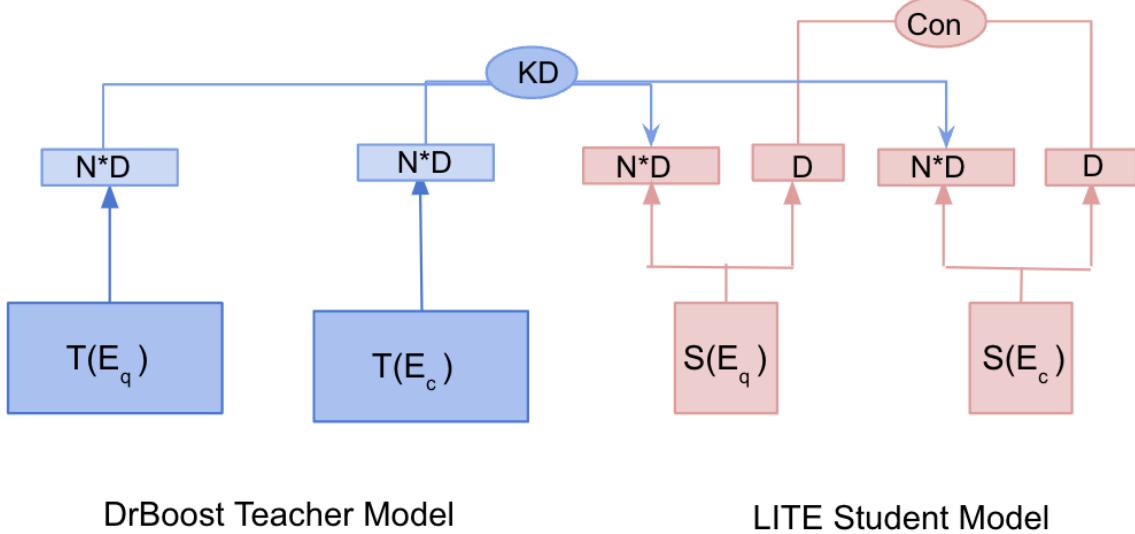


Figure 3.1: The teacher model (DrBoost) consists of N weak-learners and produces embeddings of dimension $N*D$. The student model (LITE) has one weak-learner and produces two embeddings: one has dimension of D , and one has dimension of $N*D$. The smaller embeddings learn to maximize the similarity between question and positive context embeddings, and the larger embeddings learn the embeddings from the teacher model.

We introduce a LITE model that further reduces the memory of DrBoost, which is jointly trained on contrastive learning and knowledge distillation from DrBoost (see Figure 3.1). We then integrate BM25 with either LITE and DrBoost to form light hybrid retrievers (Hybrid-LITE and Hybrid-DrBoost) to assess whether light hybrid retrievers can achieve both memory-efficiency and sufficient performance.

We conduct experiments on the NaturalQuestion dataset (Kwiatkowski *et al.*, 2019b) and draw interesting results. First of all, LITE retriever maintains 98.7% of the teacher model performance and reduces its memory by 2 times. Second, our Hybrid-LITE saves more than $13\times$ memory compared to Hybrid-DPR, while maintaining more than 98.0% performance; and Hybrid-DrBoost reduces the indexing memory ($8\times$) compared to Hybrid-DPR and maintains at least 98.5% of the performance. This shows that the light hybrid model can achieve sufficient performance while reducing

the indexing memory significantly, which suggests the practical usage of light retrievers for memory-limited applications, such as on-devices.

One important reason for using hybrid retrievers in real-world applications is the generalization. Thus, we further study if reducing the indexing memory will hamper the generalization of light hybrid retrievers. Two prominent ideas have emerged to test generalization: out-of-domain (OOD) generalization and adversarial robustness (Gokhale *et al.*, 2022). We study OOD generalization of retrievers on EntityQuestion (Sciavolino *et al.*, 2021). To study the robustness, we leverage six techniques (Morris *et al.*, 2020) to create adversarial attack testing sets based on NQ dataset. Our experiments demonstrate that Hybrid-LITE and Hybrid-DrBoost achieve better generalization performance than individual components. The study of robustness shows that hybrid retrievers are always better than sparse and dense retrievers. Nevertheless all retrievers are vulnerable, suggesting room for improving the robustness of retrievers, and our datasets can aid the future research.

3.1 Related Work

Hybrid Retriever integrates the sparse and dense retriever and ranks the documents by interpolating the relevance score from each retriever. The most popular way to obtain the hybrid ranking is applying linear combination of the sparse/dense retriever scores (Karpukhin *et al.*, 2020a; Ma *et al.*, 2020; Luan *et al.*, 2021; Ma *et al.*, 2021d; Luo *et al.*, 2022c). Instead of using the scores, Chen *et al.* (2022a) adopts Reciprocal Rank Fusion (Cormack *et al.*, 2009) to obtain the final ranking by the ranking positions of each candidate retrieved by individual retriever. Arabzadeh *et al.* (2021) trains a classification model to select one of the retrieval strategies: sparse, dense or hybrid model. Most of the hybrid models rely on heavy dense retrievers, and one exception is (Ma *et al.*, 2021d), where they use linear projection, PCA, and product

quantization (Jegou *et al.*, 2010) to compress the dense retriever component. Our hybrid retrievers use DrBoost as the dense retriever, which is more memory-efficient and achieves better performance than the methods used in (Ma *et al.*, 2021d). More importantly, we introduce a LITE dense retriever to further reduce the indexing memory.

Indexing-Efficient Dense Retriever Most of the existing dense retrievers are indexing heavy (Karpukhin *et al.*, 2020a; Khattab and Zaharia, 2020; Lee *et al.*, 2021; Luo, 2022). To improve the indexing efficiency, there are mainly three types of techniques. One is to use vector product quantization (Jegou *et al.*, 2010). Second is to compress a high dimension dense vector to a low dimension dense vector, for e.g. from 768 to 32 dimension (Lewis *et al.*, 2021a; Ma *et al.*, 2021d). The third way is to use a binary vector. BPR (Yamada *et al.*, 2021) and JPQ (Zhan *et al.*, 2021) produce the document indexing by 768 dimension binary vectors.

Generalization of IR Two main benchmarks have been proposed to study the OOD generalization of retrievers, BEIR (Thakur *et al.*, 2021a) and EntityQuestion (Sciavolino *et al.*, 2021). As shown by previous work (Thakur *et al.*, 2021a; Chen *et al.*, 2022a), the generalization is one major concern of DR. To address this limitation, Wang *et al.* (2021) proposed GPL, a domain adaptation technique to generate synthetic question-answer pairs in specific domains. A follow-up work Thakur *et al.* (2022) trains BPR and JPQ on the GPL synthetic data to achieve efficiency and generalization. Chen *et al.* (2022a) investigates a hybrid model in the OOD setting, yet different from us, they use a heavy DR and do not concern the indexing memory. Most existing work studies OOD generalization, and much less attention paid toward the robustness of retrievers (Penha *et al.*, 2022; Zhuang and Zuccon, 2022; Chen *et al.*, 2022b). To

study robustness, Penha *et al.* (2022) identifies four ways to change the syntax of the queries but not the semantics. Our work is a complementary to Penha *et al.* (2022), where we leverage adversarial attack techniques (Morris *et al.*, 2020) to create six different testing sets for NQ dataset (Kwiatkowski *et al.*, 2019b).

3.2 Model

3.2.1 Preliminary

BM25 Robertson *et al.* (2009), is a bag-of-words ranking function that scores the query (Q) and document (D) based on the term frequency. The following equation is the one of the most prominent instantiations of the function,

$$score(D, Q) = \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgdl})}, \quad (3.1)$$

where $\text{IDF}(q_i)$ is the inverse document frequency of query term q_i , $f(q_i, D)$ is the frequency of q_i in document D , $|D|$ is the length of the document D , and $avgdl$ is the average length of all documents in the corpus. In practice, $k_1 \in [1.2, 2.0]$ and $b = 0.75$. BM25 is an unsupervised method that generalizes well in different domains (Thakur *et al.*, 2021b).

DPR Dense passage retriever involves two encoders: the question encoder E_q produces a dense vector representation V_q for an input question q , and the context encoder E_c produces a dense vector V_c representation for an input context c . Both encoders are BERT model and the output vector are the embeddings of the special token [CLS] in front of the input text (Eq. 3.2).

$$V_q = E_q(q) [\text{CLS}], \quad V_c = E_c(c) [\text{CLS}]. \quad (3.2)$$

The score of c w.r.t q is the inner-dot product of their representations (Eq 3.3).

$$\text{sim}(q, c) = \mathbf{V}_q^\top \mathbf{V}_c. \quad (3.3)$$

DPR uses contrastive loss to optimize the model such that the score of positive context c^+ is higher than the score of the negative context c^- . Mathematically, DPR maximizes the following objective function,

$$\mathcal{L}_{con} = -\log \frac{e^{\text{sim}(q, c^+)}}{e^{\text{sim}(q, c^+)} + \sum_{j=1}^n e^{\text{sim}(q, c_j^-)}}, \quad (3.4)$$

where n is the number of negative contexts. For better representation learning, DPR uses BM25 to mine the hard negative context and the in-batch negative context to train the model.

DrBoost is based on ensemble learning to form a strong learner by a sequence of weak learners, and each weak learner is trained to minimize the mistakes of the combination of the previous learners. The weak learner has the similar architecture as DPR Karpukhin *et al.* (2020a), but the output vectors are compressed to a much lower dimension by a linear regression layer \mathbf{W} ,

$$\mathbf{v}_q^i = \mathbf{W}_q \cdot \mathbf{V}_q^i, \quad \mathbf{v}_c^i = \mathbf{W}_c \cdot \mathbf{V}_c^i, \quad (3.5)$$

where $\mathbf{V}_{q/c}^i$ are the representation of question/document given by the embeddings of special tokens [CLS] of a high dimension, $\mathbf{v}_{q/c}^i$ are the lower embeddings produced by the i^{th} weak learner. The final output representation of DrBoost is the concatenation of each weak learners' representations as expressed by Eq. 3.6.

$$\mathbf{q} = [\mathbf{v}_q^1, \dots, \mathbf{v}_q^n], \quad \mathbf{c} = [\mathbf{v}_c^1, \dots, \mathbf{v}_c^n], \quad (3.6)$$

where n is the total number of weak learners in the DrBoost. The training objective of DrBoost is

$$\mathcal{L}_{con} = -\log \frac{e^{\text{sim}(q, c^+)}}{e^{\text{sim}(q, c^+)} + \sum_{j=1}^{j=n} e^{\text{sim}(q, c_j^-)}}, \quad (3.7)$$

where $\text{sim}(q, c)$ is the inner-dot product.

3.2.2 LITE: A Light Dense Retriever

Since DrBoost has N encoders, Lewis *et al.* (2021a) distills the knowledge of multiple encoders to a single encoder to save inference time. In consequence, the student model has one encoder which produces the same indexing memory as the teacher model. Here, we want to further reduce the student indexing memory. To achieve this, we introduce a LITE retriever (see Figure 3.1), which produces two embeddings for an input text: one has a smaller dimension ($v_{q/c,s}$) for retrieval task, and the other one is a larger dimension ($v_{q/c,l}$) for learning knowledge from a teacher model. These embeddings are obtained by compressed the [CLS] token by separate linear regression layers, mathematically,

$$v_{q/c,s} = W_{q/c,s} \cdot V_{q/c}, \quad v_{q/c,l} = W_{q/c,l} \cdot V_{q/c} \quad (3.8)$$

$v_{q/c,s}$ is optimized by the contrastive loss (Eq. 3.7). And $v_{q/c,l}$ learns the teacher model embeddings. The knowledge distillation (KD) loss is composed of three parts (Eq. 3.9): 1) the distance between student question embeddings and the teacher question embeddings, 2) the distance between student context embeddings and the teacher context embeddings, and 3) the distance between student question embeddings and the teacher positive context embeddings.

$$\mathcal{L}_{KD} = \|v_{q,l} - \mathbf{q}\|^2 + \|v_{c,l} - \mathbf{c}\|^2 + \|v_{q,l} - \mathbf{c}^+\|^2 \quad (3.9)$$

The final objective of the student model is,

$$\mathcal{L}_{joint} = \mathcal{L}_{con} + \mathcal{L}_{KD}. \quad (3.10)$$

After LITE is trained, $\mathbf{v}_{c,s}$ is used to indexing the context and thus save the memory compared to the distilled model in Lewis *et al.* (2021a).

3.2.3 Memory Efficient Hybrid Model

Most of the existing hybrid models combine BM25 with DPR (or similar variants) that aims to improve model performance on domains same as the training sets. Although such hybrid models achieve better performance, they come at the cost of larger memory. Different from previous hybrid models, we combine BM25 and DrBoost, which as described in previous section, largely reduce the indexing memory of DPR, and thus yields hybrid models with less memory compared to previous hybrid models.

Our hybrid models retrieve the final documents in a re-ranking manner. We first retrieve the top-k candidate articles using BM25 and DrBoost separately. The scores produced by these two methods for each candidate are denoted by S_{BM25} and S_{DR} respectively and normalized to the $[0, 1]$ range by MinMax normalization to obtain S'_{BM25} and S'_{DR} . If a candidate article is not retrieved by either retriever, then its score for that retriever is 0. For each article, we get a new score:

$$S_{\text{hybrid}} = w_1 \times S'_{\text{BM25}} + w_2 \times S'_{\text{DR}}, \quad (3.11)$$

where w_1 and w_2 denote the weights of BM25 and DrBoost scores respectively. The weights can be learned from the development set if it is available. In our experiments, we simply set equal weights (i.e. 0.5) to each method. Finally, we re-rank candidates based on S_{hybrid} and return the top-k candidates.

3.3 Datasets to Probe Robustness of IR Systems

Unlike out-of-domain generalization, where a model is tested on datasets from different domains w.r.t the training sets, robustness refers to the performance of a model on adversarial attack samples obtained by small perturbations of the original dataset. The distribution of the attack samples are shifted from the original dataset within a boundary. Robustness is an important feature of IR systems since the information needs of queries can be expressed in different ways. To test the robustness of IR systems, we take NQ (Kwiatkowski *et al.*, 2019b) as the original dataset, and create 6 different adversarial attack samples from TextAttack library (Morris *et al.*, 2020). Each method is chosen because they do not change the original meaning of the queries and the relevant documents should be the same as the original relevant documents. Figure 3.2 shows an example obtained by each method.

Char-Swap (CS): augments words by swapping characters out for other characters. There are four transformations, and for each query from the original dataset, we randomly choose one: swap two adjacent letters in the word; substitute a letter in the word with a random letter; delete a random letter from the word; and insert a random letter in the word. **Word Deletion (WD):** delete a word randomly from the original query. **Synonym Replacement (SR):** replaces a word in the query with synonym from the WordNet (Miller, 1992). **Word-Order-Swap (WOS):** swaps the order of the words in the original query. **Synonym Insertion (SI):** insert a synonym of a word from the WordNet to the original query. **Back-Translation** translates the original query into a target language and translates it back to the source language. We use the machine translation model from (Tiedemann and Thottingal, 2020) and the target language is Spanish and the source language is English.

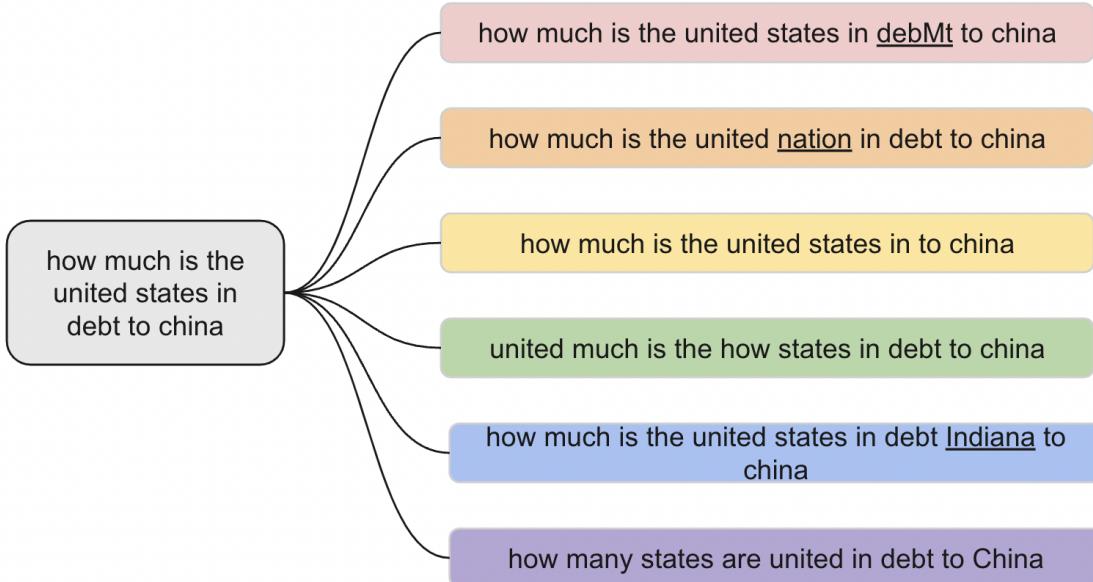


Figure 3.2: Examples of the adversarial attack questions. Underline denotes the change from the original question. The example from the top to the bottom are augmented by CS, WD, SR, WOS, SI, and BT.

3.4 Experiments and Results

Existing Methods We include four existing methods for comparison in this work, DrBoost (Lewis *et al.*, 2021a), DPR (Karpukhin *et al.*, 2020a), SPAR (Chen *et al.*, 2021) and hybrid model BM25 + DPR (Karpukhin *et al.*, 2020a). SPAR is an ensemble model that involves two encoders, which achieves the state-of-the-art (SOTA) performance on the NQ dataset. We refer reader to the details of SPAR in the original paper. We report the numbers of DrBoost from the original paper and the rest three methods from (Chen *et al.*, 2021).

Baselines We present three baselines result, BM25, DPR₃₂, and DrBoost-2. DPR₃₂ refers to DPR with a linear project layer of dimension 32. We take DPR₃₂ as the first weak learner, and use it to mine negative passages to train the next stage weak

learner¹, and then combine these two models to form DrBoost-2 which produce 64 dimension vectors. Other details of the implementation is given in Appendix. We do not go beyond 2 weak learners because our goal is to achieve memory-efficiency while increasing the number of encoders in the DrBoost will yield larger indexing.

Our Models are mainly LITE and the three light hybrid models. LITE is trained by the method we introduce in §3.2.2 with the knowledge from DrBoost-2 teacher model. We present three hybrid models BM25 + LITE, BM25 + DPR₃₂, and BM25 + DrBoost-2, which are memory-efficient compared to existing methods. Next we present the experiments and the findings.

3.4.1 Memory Efficiency and Performance

LITE achieves much better performance compared to DPR₃₂, while both use the same amount of memory. LITE also maintains more than 98% knowledge of its teacher (DrBoost-2), and importantly save 2 times of indexing memory. These demonstrates the effectiveness of our proposed training technique.

Hybrid-LITE achieves better performance than DrBoost-2 while using less indexing memory. Hybrid-LITE also matches the performance of DrBoost in terms of R@100 (87.4 v.s. 87.2) while uses 3 times less memory. Comparing with the hybrid model BM25 + DPR, Hybrid-LITE maintains 98.4% performance but uses 13 times less memory. Comparing with the SOTA model SPAR, Hybrid-LITE achieves its 98.2% performance and uses 25 times less memory.

¹We use in-batch negative with one hard negative example in the training time, while in the original DrBoost paper, they do not use in-batch negative for training.

Hybrid-DrBoost-2 achieves comparable performance as DrBoost (0.9 worse on R@20 less but same on R@100), and uses 2 times less memory of DrBoost. This shows the effects of BM25 match the capacity of 4 encoders in the DrBoost (which uses 6 encoders). We also compare Hybrid DrBoost-2 with BM25 + DPR or SPAR, where our model achieves almost 99% performance but uses less than 8 times or 16 times of memory.

Method	Index-M (GB)	NQ		EntityQuestion	
		R@20	R@100	R@20	R@100
Existing Method					
DrBoost	15.4/13.5	81.3	87.4	51.2	63.4
DPR	61.5	79.5	86.1	56.6	70.1
BPR	2	77.9	85.7	-	-
BM25+DPR	63.9	82.6	88.6	73.3	82.3
SPAR	123.0	83.6	88.8	74.0	82.0
Our Baseline					
BM25	2.4	63.9	78.8	71.2	79.7
DPR ₃₂	2.5	70.4	80.0	31.1	45.5
DrBoost-2	5.1	77.3	84.5	41.3	54.2
Our Model					
Lighter	2.5	75.1	83.4	35.0	48.1
Hybrid-Lighter	4.9	79.9	87.2	71.5	80.8
Hybrid-DPR ₃₂	4.9	77.7	86.2	70.8	80.5
Hybrid-DrBoost-2	7.5	80.4	87.5	72.4	81.4

Table 3.1: Performance of existing methods, our baselines and our hybrid model on NQ dataset. The performance of DrBoost on NQ is using 6 weak learners (15.4 GB indexing memory) and of EntityQuestion is using 5 weak learners (13.5 GB).

3.4.2 Generalization Results

In Domain Generalization We study the generalization of our models on EntityQuestion dataset, which is considered as in-domain generalization because the question and corpus of EQ is from the same domain as the training set. EQ has been shown to be very difficult for dense retriever but easy for BM25 (Sciavolino *et al.*, 2021). Indeed, our dense retriever results are align with the previous finding that the performance of DPR₃₂ and DrBoost-2, and LITE are worse than BM25. Nevertheless, our hybrid models improves both BM25 and the DrBoost performance and our Light hybrid model achieves similar performance as hybrid of DPR and SPAR while has large advantage in terms of the memory as we described in previous section.

Out-of Domain Generalization We study the out-of-domain generalization where the question and corpus are from different domains as the training set. We study BEIR benchmark (Thakur *et al.*, 2021a) and use the MS MARCO dataset to train the dense retrievers, and evaluate on 13 public available datasets in BEIR. We present the results of our two dense retrievers and two hybrid retrievers along with BM25. We take BM25 as our major baseline for its efficient indexing memory and strong performance, while others (e.g. SPAR) improves upon BM25 but come at the large cost of indexing memory. Also, for simplicity, we do not show the results of DPR₃₂ as demonstrated by previous section, the performance of which is worse than LITE. Table 3.2 shows that two of our dense retrievers do not have good generalization, align with the findings in (Thakur *et al.*, 2021a). However, our LITE can outperform DrBoost on two datasets (Touche-2020 and DBpedia). Comparing with BM25, both Hybrid-DrBoost-2 and Hybrid-LITE achieve better performance on 6 datasets, while Hybrid-DrBoost-2 achieve the best performance on 5 datasets and Hybrid-LITE is the best on 1 dataset. Nevertheless, we see that hybrid retrievers do not always improve

the BM25 performance. One potential way to improve the performance of hybrid models is to assign different weights to each retriever. We learn the weights using the development set of MS MARCO. However, the best pair of weights of learned from in-domain data do not yield better performance on out-of-domain (the result is given in Appendix). We leave how to assign different weights to hybrid models as a future work.

Method(→)	BM25	LITE	DrBoost-2	Hybrid-2	Hybrid-L
MS MARCO	0.260	0.329	0.341	0.373	0.363
TREC-COVID	0.632	0.446	0.474	0.603	0.579
NFCorpus	0.322	0.208	0.211	0.310	0.305
NQ	0.306	0.325	0.349	0.423	0.406
HotpotQA	0.633	0.201	0.232	0.564	0.549
FiQA-2018	0.236	0.142	0.148	0.249	0.241
ArguAna	0.397	0.286	0.306	0.364	0.350
Touché-2020	0.442	0.267	0.250	0.375	0.391
Quora	0.787	0.378	0.387	0.775	0.747
DBPedia	0.318	0.243	0.236	0.344	0.346
SCIDOCs	0.149	0.070	0.075	0.135	0.133
FEVER	0.678	0.473	0.498	0.735	0.724
Climate-FEVER	0.165	0.137	0.143	0.220	0.214
SciFact	0.707	0.307	0.330	0.651	0.648
Average	0.431	0.272	0.284	0.430	0.435

Table 3.2: Performance of light retrievers on BEIR in terms of nDCG@10. MS MARCO is evaluated on the Dev set. Hybrid-2: Hybrid-DrBoost-2, Hybrid-L: Hybrid-LITE.

3.4.3 Robustness Results

We compare the robustness of each model in terms of both performance (the higher R@K a model achieves, the more robust the model is) and the average drop w.r.t the original performance on NQ dataset (i.e. the smaller drop of the model achieves, the more robust the model is).

First of all, from Table 3.3 we observe that all models perform worse across most of the adversarial attack dataests (5 out of 6) compared to the original performance, which showcase that the current retrievers are not robust enough. Interestingly, Table 3.3 shows that both dense retriever and sparse retriever are quite robust on word-order-swap (WOS) adversarial questions. It is expected that BM25 will be robust on this type of questions, yet it is not straightforward that dense retriever is also robust on this type of question. This shows that the order of the words in the question is not important for the dense retriever neither. We also see that char-swap (CS) is the most difficult type of questions among all adversarial questions, which means that both type of retrievers might not perform well when there are typos in the questions.

Diving into the individual performance of each retriever, we see that some models are more robust than others. For example, in Table 3.3, we compare LITE with DPR₃₂ given that these two models have the same indexing memory and find that LITE is more robust. We also compare the hybrid model with the pure dense retriever counterparts (e.g. compare hybrid Drboost-2 with DrBoost-2), and find that hybrid models are consistently more robust. This suggests that the hybrid model can mitigate the performance drop of both BM25 and dense retriever.

Method	R@100							
	Ori	CS	WD	SR	WOR	SI	BT	Drop
BM25	78.8	68.2	71.7	74.5	78.3	77.2	71.2	5.9
LITE	83.4	69.3	71.8	78.9	81.2	79.0	75.6	7.9
DPR ₃₂	80.8	61.9	65.8	75.3	76.4	73.3	71.1	10.3
DrBoost-2	84.5	71.6	80.1	74.7	82.6	80.4	77.9	7.8
DPR ₇₆₈	86.1	74.8	78.9	82.5	85.0	83.4	80.3	5.5
+LITE	83.4	76.5	78.0	83.7	86.6	85.4	80.8	5.1
+DPR ₃₂	86.2	74.4	78.0	82.7	84.9	83.2	78.6	6.1
+DrBoost-2	87.5	77.7	84.6	81.0	86.7	85.9	81.9	5.2
+DPR ₇₆₈	88.3	78.6	82.9	85.4	87.7	86.6	82.6	4.4

Table 3.3: Ori: Original question; CS: CharSwap; WD:Word deletion; WSR: WordNet synonym replacement; WOR: Word order swaps; RSI :Random synonym insertion; BT: Back Translation. The smaller the Average Drop is, the more robust the model is.

3.4.4 Ablation Study

LITE Can Improve DrBoost Recall that DPR₃₂ is one encoder in DrBoost-2, and since LITE performs better than DPR₃₂ (see Table 3.1), we ask the question can LITE replaces DPR₃₂ to form a stronger DrBoost-2 model? To answer this question, we compare the performance of R-DrBoost-2 (i.e. replace DPR₃₂ with LITE) with the original DrBoost-2. From Table 3.4, We observe that R-DrBoost-2 performs worse than DrBoost-2, indicating that the encoders in the DrBoost indeed relate and complement to each other and replacing an unrelated encoder degrades the performance. Then we ask another question, can we train a weak learner that minimizes the error of

LITE, and combine LITE with the new weak learner to form a stronger DrBoost (L-DrBoost-2)? Table 3.4 shows L-DrBoost-2 is better than DrBoost-2, and hybrid L-DrBoost-2 is better than hybrid DrBoost-2 as well (81.0 v.s. 80.4 on R@20). This indicates that starting with a stronger weak learner can yield a stronger DrBoost.

Metric	O-DrBoost	R-DrBoost	LITE-DrBoost	H-LITE-DrBoost
R@20	77.3	75.6	77.9	81.0
R@100	84.5	83.9	84.7	87.5

Table 3.4: Three DrBoost (with 2 weak learners) and one hybrid retriever. O-DrBoost: the original DrBoost, R-DrBoost: replace the first weak learner in O-DrBoost with LITE, LITE-DrBoost: use LITE as the first weak learner and mine negative using LITE to train a new weak learner to form a DrBoost, H-LITE-DrBoost: hybrid BM25 with LITE-DrBoost.

Hybrid model consistently improves the DrBoost performance. We study six DrBoost models with 1-6 weak learners. In Figure 3.3, we see that the performance of hybrid models consistently improves the DrBoost performance, demonstrating the results of BM25 and DrBoost complement each other and combining two models improves individual performance. We also see that the improvement is larger when the DrBoost is weaker, e.g. hybrid model significantly improves DPR₃₂.

Different Hybrid Scores In our hybrid model, besides the hybrid scores we introduced in §3.2.3, we also study two different hybrid scores of BM25 and the DrBoost. Simple Summation is to add two scores together, and multiplication is to mutiply two scores. We compare two hybrid models’ performance, Hybrid-DrBoost-2 and Hybrid-DrBoost-6. Table 3.5 shows that the MinMax normalization performs the best (except that simple summation is slightly better in terms of R@20 for hybrid models with 6 weak learners).

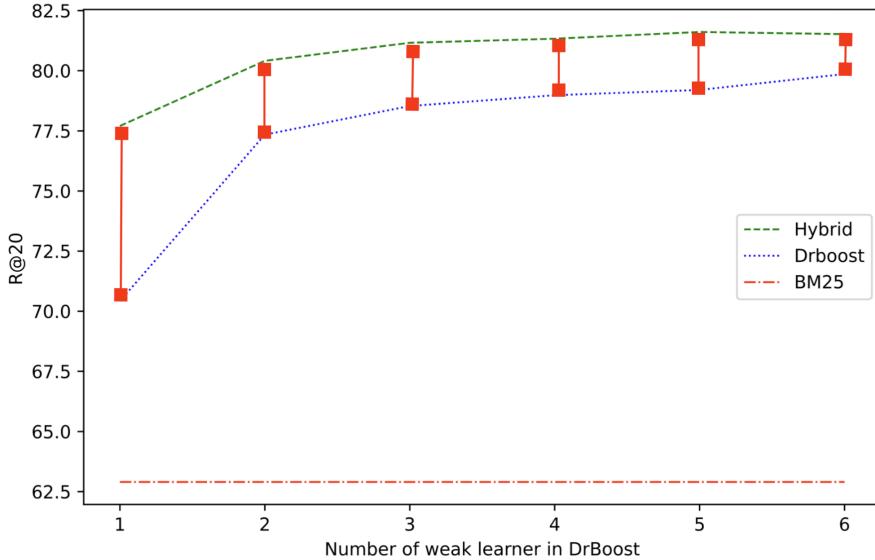


Figure 3.3: Compare DrBoost, BM25 and the Hybrid models performance.

3.5 Discussion and Summary

To achieve indexing efficiency, in this work, we study light hybrid retrievers. We introduce LITE, which is jointly trained on contrastive learning and knowledge distillation from the state-of-the-art indexing-efficient dense retriever, DrBoost. Then, we integrate BM25 with LITE or DrBoost to form light hybrid retrievers. Our light hybrid models achieve sufficient performance and largely reduce memory. We also study the generalization of retrievers and suggest that all sparse, dense, and hybrid retrievers are not robust enough, which opens up a new avenue for research. In future, there are few directions, we can improve beyond current work.

Replace BM25 with Dense Retriever While our hybrid retrievers save memory indexing compared to existing hybrid retrievers, they require separate retrieval because The sparse and dense retrievers use different indexing methods, which increases the inference complexity of the hybrid models. A potential method to overcome this issue is SPAR-like approach where a dense retriever model is used to mimic the BM25

Model	Method	NQ	
		R20	R100
Hybrid(32*2)	Simple Sum	79.03	84.63
	Multiplication	79.03	84.63
	MinMax and Sum	80.41	87.47
Hybrid(32*6)	Simple Sum	81.61	86.12
	Multiplication	81.19	86.12
	MinMax and Sum	81.52	88.28

Table 3.5: Compare three hybrid scores. We study two hybrid model, BM25 with 2 weak learners (32*2) and BM25 with 6 weak learners (32*6)

behavior so that the hybrid model can be the concatenation of two dense retrievers. Such a model is termed as Lambda model. Following Luan *et al.* (2021), we use BM25 to generate the training data and train 6 Lambda models that project vectors of 768 dimension to 32/64/96/128/160/192, respectively. However, our Lambda models with lower dimensions can not imitate BM25 well and perform worse than BM25. Replacing BM25 with low capacity Lambda models will degrade the hybrid models performance. We leave how to sufficiently distill BM25 knowledge to a low dimension Lambda model as a future work.

Replace DrBoost with other efficient dense retriever As we mention in the related work, there are other options of efficient DR, for example, BPR, which uses binary vectors to represent the queries and documents. However, BPR itself already involves a re-ranking stage which will make the hybrid-BPR more complex. It will be an interesting future work to explore if the hybrid model of BM25 and BPR can achieve reasonable performance without the re-ranking stage in BPR.

Chapter 4

NEURAL-RERANKER

Due to the wide applications in real-world, Retrieval Based Question Answering (ReQA) has gained increasing interest and attention in recent years, and many benchmarks have been proposed for Retrieval Based Question Answering (ReQA) (Cohen *et al.*, 2018; Khot *et al.*, 2020; Ahmad *et al.*, 2019; Guo *et al.*, 2021). A promising approach for solving ReQA involves two stages: retrieve a small set of candidates from a large corpus and re-rank these candidates. The re-ranking stage can significantly improve the initial retrieval performance (Ozyurt *et al.*, 2020), and thus it is crucial for any retrieval system (Ma *et al.*, 2021a).

Large Pretrained Language Models (PrLMs) have been widely used as neural re-rankers (Yilmaz *et al.*, 2019; Nogueira and Cho, 2019). In most cases, the negative examples used to train the re-ranker are assigned the same label. However, we argue that some candidates may be more negative than others and should be treated differently. Figure 4.1 shows an example from the HotpotQA dataset (Yang *et al.*, 2018b) to illustrate this argument. In this example, neither S1 nor S2 contains the correct answer; yet S1 mentions a key entity in the question (David Beckham), while S2 has no common entity with the question. From the human perspective, S1 should have a higher score than S2.

It leads us to ask a question - “is having different levels of negativeness beneficial for training neural re-rankers?” Driven by this question, we propose an approach for scoring negative candidates (§4.2). Our approach has two stages. First, we train a model on STS benchmark (Conneau and Kiela, 2018). This model generates a

Question: The manager who recruited David Beckham managed Manchester United during what time frame?

Answer: Sir Alexander Chapman Ferguson, CBE (born 31 December 1941) is a Scottish former football manager and player who managed Manchester United **from 1986 to 2013**.

S1: Instead, he had drafted in young players like Nicky Butt, David Beckham, Paul Scholes and the Neville brothers, Gary and Phil.

S2: The awards ceremony was held at Earls Court in London for the last time.

Figure 4.1: An example from the HotpotQA dataset. While both S1 and S2 are negative candidates to the question, our approach assigns a higher negativeness score to S1 than S2.

high score for a two-sentence pair if they are semantically similar; otherwise, a low score. Second, we use this STS model to generate scores for the question and negative candidate pairs. In this way, we obtain a set of question-candidate pairs with labels in the continuous range of [0, 5] as opposed to previous works where labels are binary. Furthermore, we want the generated score for a negative candidate to be higher than others if the first candidate has more relevant information to the answer. To achieve this goal, we explore three data augmentation techniques (§4.2.2). Such a scoring approach allows: 1) a good candidate that is not annotated as an answer to have a high score, 2) more negative samples to be used to train a neural re-ranker, and 3) negative candidates to be differentiated using “negativeness” scores. In this paper, the negativeness score means the score for a question and a negative candidate; and a higher score means the negative candidate contains more information to answer the question.

We compare three standard training strategies and our proposed method on the MultiReQA (Guo *et al.*, 2021) benchmark, which includes five training datasets across different domains. Based on our experiments, we observe that 1) our proposed approach outperforms three baselines by up to 13% absolute improvement on the

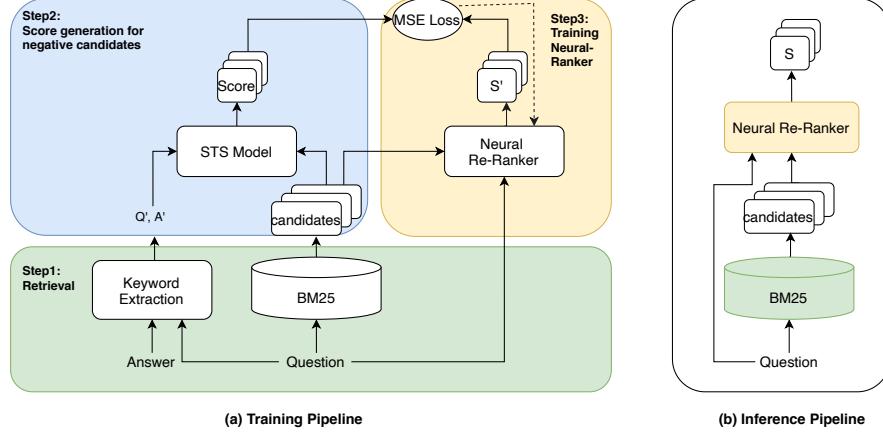


Figure 4.2: (a) Training Pipeline: Step1—retrieve negative candidates for a question using BM25; Step2—use a frozen STS model to generate negativeness scores for a question and candidate pair; and Step3—train a neural re-ranker using the generated scores given by the STS model. (b) Inference Pipeline: retrieve the top 100 candidates using BM25 and re-rank them using a neural re-ranker. Q' and A' means augmented questions and answers, and S' means predicted scores of neural re-ranker.

SearchQA dataset and 5.5% on average across all datasets in terms of P@1; 2) use of a different negativeness score achieves better performance than the same score even when fewer negative candidates are used; and 3) our proposed method has a significant advantage in a low-resource setting. These lead to the answer to the question that the use of a negativeness score is an efficient way to train a neural re-ranker.

4.1 Related Work

Retrieval Based Question Answering ReQA is to identify sentences from a large corpus that contain the answer to a question (Yang *et al.*, 2015; Cakaloglu *et al.*, 2020; Ahmad *et al.*, 2019; Guo *et al.*, 2021). It has practical applications such as Google Talk to Books¹. ReQA is similar to Open Domain Question Answering (ODQA) but different in the following aspect, ReQA aims to build an efficient retrieval system, and the answer is a sentence or a short passage (Ahmad *et al.*, 2019); while ODQA requires a retrieval system to find relevant documents at a large scale and a machine reading

¹<https://books.google.com/talktobooks/>

comprehension model to predict short answer span from documents (Bilotti *et al.*, 2007; Chen and Van Durme, 2017; Chen *et al.*, 2017a; Min *et al.*, 2019; Karpukhin *et al.*, 2020b). In this paper, we focus on the ReQA task and believe that building an efficient system for ReQA is also beneficial for the ODQA task. For example, QASC (Khot *et al.*, 2020) requires retrieving sentences from a large corpus and composing them to answer a multiple-choice question, and a good ReQA system can be used to retrieve sentences in the first stage.

Neural Re-Ranker Bag-of-words ranking models such as BM25 (Robertson and Zaragoza, 2009a) have a long history in information retrieval. Although efficient, these methods depend on handcrafted features and can not be optimized for a specific task such as ReQA. Therefore, a re-ranker is trained on a downstream task to re-score the candidates after the first step of retrieval. Neural networks have been applied as re-rankers (Guo *et al.*, 2016; Hui *et al.*, 2017; Xiong *et al.*, 2017; Dai *et al.*, 2018; McDonald *et al.*, 2018), also called as answer selection models in some work (Rao *et al.*, 2016; Yang *et al.*, 2015; Rao *et al.*, 2019; Laskar *et al.*, 2020). Boosting technique has been proposed to train a neural re-ranker where the training samples are assigned with different weightsMakino and Iwakura (2017). Recently, large language models like BERT (Devlin *et al.*, 2019a) and RoBERTa (Liu *et al.*, 2019a) are widely used as re-rankers (Nogueira and Cho, 2019; Yilmaz *et al.*, 2019; MacAvaney *et al.*, 2019). Such re-rankers take the concatenation of a query and a candidate as input and apply attention technique(Vaswani *et al.*, 2017a) to allow rich interaction between the question and the candidate. Then a classification or regression module (scoring layer) is added on top to compute a score. Binary classification entropy (BCE) is usually used to train a re-ranker, but BCE has limitations such as a large number of negative candidates being unused to create balanced training samples. Triplet loss addresses

this issue by the idea of learning to rank Liu (2009). However, none of these methods addresses the concern of whether we can use different negativeness scores to train a neural re-ranker. Similar to previous work, we use large PrLMs as re-rankers but different from theirs, we train a model using different scores for negative candidates. Re-ranking using ensemble models has been explored recently Zhang *et al.* (2021), but since their systems are more complex than ours in terms of model size, we don't compare with them.

4.2 Negative Candidate Scoring Approach

The key idea of the negative candidate scoring approach is to utilize a Semantic Textual Similarity (STS) model and the motivation is that the STS score determines how close two sentences are in terms of semantic meaning (Conneau and Kiela, 2018).

Review STS STS determines how close two sentences are in terms of semantic meaning (Conneau and Kiela, 2018). Specifically, given two sentences, a high STS score indicates that they present similar meanings; while a low score implies that they have different meanings. The STS score is in the range of [0, 5]. Table 4.1 shows two pairs of sentences with scores 0 and 5 from the STS-B dataset. Score 5 means two sentences are semantically equivalent and score 0 means semantically irrelevant.

Sentence 1	Sentence 2	Score
A man is playing a guitar.	A man plays the guitar.	5.0
A young man is playing the piano.	A woman is peeling a prawn.	0.0

Table 4.1: Two examples from the STS-benchmark, the first pair of sentences have the highest score since they are highly similar, while the second pair has the lowest score because they have totally different meanings.

Motivation of Using STS to Generate Scores STS lays the foundation of our scoring approach because there is a relation between the STS task and the question-candidate ranking task. Considering a question and a candidate pair, if the candidate has similar information regarding the question, then it is likely to be a relevant candidate (corresponding to a high STS score); on the contrary, if it has less similar information, then it is likely to be irrelevant (corresponding to a lower STS score). Meanwhile, STS is better than other methods of finding similar information because it considers the semantic meaning of two sentences such as synonyms of words.

In the following, we describe the two stages of our scoring approach: (1) training an STS model, and (2) using it to generate negativeness scores for the question and negative candidate pairs.

4.2.1 Training an STS-model

We train an STS model on the STS benchmark, which is a regression model consisting of a RoBERTa model (Liu *et al.*, 2019a) and a Multi-Layer Perceptron (MLP) layer. In particular, the input to the RoBERTa model is [CLS] sentence1 [SEP] sentence2 [SEP]. Then we feed the representation of the [CLS] token to the MLP layer which predicts a score. Mean Squared Error (MSE) loss is taken as the training objective to minimize the gap between the predicted score with the ground truth STS score.

4.2.2 Negativeness Score Generation

We use the STS model to generate scores for the question and negative candidate pairs. Due to the fact that sometimes, the important information is only presented in the answer but not in the question, even though a candidate is relevant to a question, the STS model might not produce a high score. To overcome this issue, we introduce

three ways to augment a question to consider the answer in the scoring process. We expect that if two candidates have similar information regarding a question, but one has more similar information to the answer than the other, then the first one should obtain a higher score. Next, we present each augmentation approach.

Question + Answer (Q+A) The first approach is to simply concatenate the answer to the original question. **Question + Keywords of Answer (Q+KA)** The second approach is to extract the keywords from the answer and concatenate the keywords to the original question. We use Rapid Automatic Keyword Extraction (RAKE) (Rose *et al.*, 2010) to extract the keywords. We believe that answer might include irrelevant information and it can be removed by extracting keywords. Neglecting irrelevant information can help the STS model generate a reasonable negative score. **Keywords of Question and Answer (KQ+KA)** This method extracts the keywords not only for the answer but also for the question. We concatenate the keywords sequentially. The intuition is the same as the second approach but also applies to the question.

4.3 Experiments and Results

4.3.1 Dataset and Evaluation Metric

We conduct experiments on MultiReQA benchmark (Guo *et al.*, 2021) which includes five training datasets: SearchQA (SQA) (Dunn *et al.*, 2017), TriviaQA (TQA) (Joshi *et al.*, 2017), HotpotQA (HQA) (Yang *et al.*, 2018b), SQuAD(Rajpurkar *et al.*, 2016), and NaturalQuestions(NQ)(Kwiatkowski *et al.*, 2019c).

Precision@K P@K reveals the proportion of top-K retrieved candidates that are relevant. R@K reveals the proportion of relevant documents in the top-K retrieved candidates. In Eq 4.1, N is the number of questions, A_K is the top-K retrieved answer,

and A^* is the correct answer.

$$P@K = \frac{1}{N} \sum_i^N \frac{|A_K \cap A^*|}{K} \quad (4.1)$$

MRR The MRR score is computed as follows,

$$MRR = \frac{1}{N} \sum_i^N \frac{1}{rank_i},$$

where $rank_i$ is the rank of the first relevant answer.

4.3.2 Baselines

We compare our proposed approach with three commonly used neural model baselines: Binary Classification Model (BCM), Regression Model (RM), and Triplet Model (TM).

Binary Classification Model (BCM) We use the RoBERTa model as the encoder, which takes input as [CLS] question [SEP] candidate [SEP]. Then, we feed the vector representation of [CLS] to a linear layer with two logits as outputs: one represents the probability of candidates being irrelevant and the other represents it being relevant. We apply binary cross-entropy loss to train this model. The training data is constructed by using the positive samples for each question with label 1, and we randomly selected the same amount of negative samples from the top 100 candidates given by BM25 with label 0.

Regression Model (RM) This baseline is similar to the BCM baseline, but the linear layer only outputs one logit instead of two, thus it is a regression model rather than a binary classification model. We use MSE loss to train this model. The positive and negative samples are the same as BCM, but the positive samples have labels

of 5. We also use 1 as the label for positive samples but find that 5 yields better performance, thus we use label 5 to train all RM baselines.

Triplet Model (TM) This baseline has the identical model architecture as the RM baseline, but we use the triplet loss to train the model and in this way, more negative candidates can be used. Specifically, each training sample is a triplet, i.e., $\langle q, c^+, c^- \rangle$, where q is a question, c^+ is a positive candidate, and c^- is a negative candidate. Let $S(q, c)$ denote the score given by the model for question q and candidate c . The model is trained such that $S(q, c^+)$ is higher than $S(q, c^-)$. We use the same negative candidates to train TM and SCONER, but SCONER use generated score as labels.

4.3.3 Results and Analysis

We use two standard metrics to evaluate each model defined by MultiReQA, P@1, and MRR. We present the most insightful results and findings in this section. In the following, we mainly describe P@1, however, it is easy to see the same trend extended to MRR.

Comparison with Baselines Table 4.2 shows that SCONER outperforms all baselines across all datasets. The largest gain SCONER achieved is $\sim 13\%$, compared to BCM on SearchQA, and the largest average gain is $\sim 5.5\%$, compared to RM. While compare to the strongest baseline, i.e., TM (since TM outperforms the other two baselines), SCONER archives $\sim 2.5\%$, $\sim 4\%$, $\sim 3\%$, and $\sim 5\%$ improvement in terms of P@1 on NQ, SQuAD, HotpotQA, and SearchQA, respectively, and outperforms TM on TriviaQA by a small margin. This shows that using more negative candidates and differentiating the negative candidates are important to boost the models' performance.

Comparison with Existing Methods The existing methods on MultiReQA directly retrieve answers from the entire corpus without re-ranking. We present one term-matching (e.g. BM25) and two neural-retrieval-based methods from Guo *et al.* (2021), which are fine-tuned BERT dual encoder and USE-QA (Yang *et al.*, 2020) on each in-domain dataset. Other baselines and our model re-rank candidates after BM25 retrieval. From the results, we see that the re-ranking phase improves performance significantly. For instance, re-ranking improves P@1 at least $\sim 20\%$, $\sim 13\%$, $\sim 42\%$, $\sim 38\%$, and $\sim 20\%$ on NQ, SQuAD, HotpotQA, SearchQA, and TriviaQA, respectively, compared to BM25.

Effect of Data Augmentation We also train neural re-rankers with scores generated by the STS model without augmentation (Q model) to see the effect of augmentation. From Table 4.2, we can see that baseline outperforms the Q model in most cases. For example, the performance of Q models is worse than TM on NQ, SearchQA, and TriviaQA. Moreover, BCM and TM outperform the Q model on average. On the other hand, using augmentation methods, Q+A, Q+KA, and KQ+KA are better than Q models and outperform all baselines on average. While the three proposed data augmentation all outperform baselines, KQ+KA is the best technique not only because it achieves the best average score but also because it performs more stable across all datasets. This demonstrates the importance of incorporating the answer to the question.

Effect of Size of Negative Candidates Here, we compare using 1/3/5/7/10 negative candidates per question to train MultiReQA-SQuAD SCONER. Figure 4.3 shows the P@1 score with different numbers of negative candidates for each method. We see that for three out of four methods, 5 or 7 already yield the best

performance, which suggests that SCONER does not need to be trained with many negative candidates. In addition, we have two observations: 1) compared to 1 negative candidate per question, 5/7/10 yield better performance, and this suggests that using one negative candidate is not enough to train SCONER; 2) compared to the TM baseline, which uses 10 negative candidates per question, all four models perform better than TM even though using fewer negative candidates, this demonstrates that using different negativeness scores is an effective way to train a neural re-ranker.

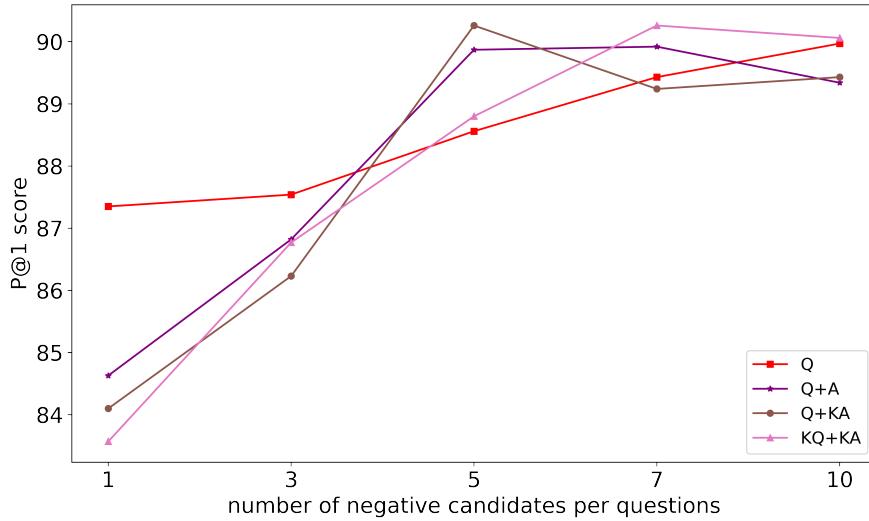


Figure 4.3: P@1 score regarding to the number of negative candidates per question used in the training. Each model is initialized with the STS model.

4.3.4 Ablation Study

What are the Effects of STS Model? STS scores can be used to approximate the scores for questions and candidates because STS and question-candidate ranking are related so these two tasks require similar knowledge or skill to solve. To further justify this intuition, we use the STS model to initialize a re-ranker rather than using the RoBERTa weights. We expect to see that the STS model will be better than the

RoBERTa model. We repeat the experiments in Table 4.2 but use the STS model rather than RoBERTa model to initialize each neural re-ranker and present the results in Table 4.3. We use green/red color to represent improvements/decrements compared to Table 4.2 (deeper color means more significant improvements/decrements). From Table 4.3, we can see that the STS model is better than the RoBERTa model in most cases, which justifies our intuition and to some extent explains why the proposed score generation approach can improve the model performance.

Can SCONER be Applied to Positive Candidates? In our previous pipeline, SCONER only uses the generated scores for negative candidates. Here, we also use the STS model to generate scores for positive candidates and use them to train the re-ranker instead of using a fixed score of 5 as in other experiments mentioned previously. We test on the SQuAD dataset. Table 4.4 shows the performance of each SCONER trained with generated scores or fixed score 5 and the best baseline model on the SQuAD dataset which is RM. We see that except for the Q model, the other three SCONERs are all better than the best baseline model, this suggests that the score for the positive candidates can be used during the training time. However, we also see that using a score of 5 is better than the generated scores in all cases. We further find that for the best SCONER model, Q+KA, 98% of the negative candidates have lower scores than the corresponding positive candidates, but the average generated score for the positive candidates is 3.64, which is less than 5. This suggests that a larger gap between scores of positive and negative candidates helps the model to differentiate the positive and negative candidates better. In addition, we also observe that the performance of the Q model, which does not use any augmentation in the generation, is much worse than the score 5 while the other three methods are similar. This suggests that the scores generated by augmentation are more reliable.

How Many Candidates are Needed for Re-ranking? To answer this question, we use 50/100/150/200 candidates in the re-ranking time. We test on three datasets using the MRR metric and consider the best model of each dataset given in Table 4.2, and they are KQ+KA, Q+KA, and Q+A models for NQ, SQuAD, and HQA, respectively. From Table 4.5, we find that the performance gap between SQuAD and NQ is more noticeable than of HotpotQA. For SQuAD, re-ranking 200 candidates yields $\sim 1\%$ improvement compared to 50, and for NQ, re-ranking 100 candidates yields $\sim 1\%$ improvement compared to 50. But for HotpotQA, re-ranking 50 candidates surprisingly yields the best performance, $\sim 0.5\%$ better than 200. Further investigation reveals that the recall of BM25 on HotpotQA is already 99% for 50 candidates and increasing the size of candidates rather introduces more distracting candidates in the re-ranking time; but for SQuAD and NQ, the recall of BM25 increases. On the other hand, re-ranking more candidates causes longer inference time, i.e. the inference time of re-ranking 50/100/150/200 candidates is 0.49/0.85/1.24/1.63 seconds. This suggests that if the recall of fewer candidates is already high enough (e.g. 99%), then using fewer candidates in re-ranking is time efficient and gets the best performance.

4.4 Discussion and Summary

We study the retrieval-based question-answering task and propose a new training strategy for a cross-attention re-ranker model. While we compare with three standard baselines and two simple retrievers, recently, there are many interesting neural retrievers have been proposed such as DPR Karpukhin *et al.* (2020b), ACNE Xiong *et al.* (2020), SPARTA Zhao *et al.* (2021), ColBERT-QA Khattab *et al.* (2021), and Poly-DPR Luo *et al.* (2022c). Comparing SCONER with these latest neural retrievers will be interesting future work. Most previous training approaches for ReQA use the same labels for all negative candidates, we argue that different candidates should

have different negativeness scores based on their semantic relevance to the question. Motivated by this, we ask the question - “can a neural re-ranker yield better performance trained on different negativeness scores?”. To answer this question, we present SCONER, a new pipeline to train neural re-rankers by generating scores for negative candidates which are based on the semantic meaning between question-candidate pairs. Our experimental results show that SCONER outperforms all standard training methods across five datasets and demonstrate that using negativeness scores to train a neural re-ranker is better than using the same labels. Our proposed method makes negative candidates differentiable which further allows us to use more negative samples to train neural re-ranker.

Metric	Model	MultiReQA					
		NQ	SQuAD	HQA	SQA	TQA	Avg.
<i>Existing Approach (without re-ranking)</i>							
	BM25	25.54	69.37	28.33	37.39	42.97	40.72
	USE-QA	38.00	66.83	31.71	31.45	32.58	40.11
	BERT	36.22	55.13	32.05	30.20	29.11	36.54
<i>Baselines</i>							
	BCM	46.07	83.71	76.60	65.48	62.05	66.78
	RM	44.76	85.36	70.61	69.79	60.41	66.19
P@1	TM	50.33	85.65	70.00	73.03	65.43	68.89
<i>SCONER (Ours)</i>							
	Q	48.64	89.09	64.76	68.64	62.20	66.67
	Q+A	49.97	89.14	79.80	70.27	64.73	70.78
	Q+KA	50.87	89.48	71.71	78.26	65.16	71.10
	KQ+KA	52.80	88.37	76.28	75.64	65.45	71.71

	<i>Existing Approach (without re-ranking)</i>						
	BM25	37.66	75.95	49.99	55.62	55.19	54.88
	USE-QA	52.27	75.86	43.77	50.70	42.39	53.00
	BERT	52.02	64.74	46.21	47.08	41.34	50.28
Metric	Model	<i>Baselines</i>					
		BCM	58.03	89.72	84.73	73.94	71.97
	RM	57.02	90.58	80.45	78.81	70.67	75.51
MRR	TM	60.87	90.27	81.00	82.22	75.30	77.93
<i>SCONER (Ours)</i>							
	Q	58.46	92.54	70.73	76.64	68.94	73.46
	Q+A	60.14	92.36	85.88	78.62	72.48	77.90
	Q+KA	60.16	92.71	80.08	84.72	72.51	78.04
	KQ+KA	61.52	91.92	82.87	83.82	73.54	78.87

Metric	Model	MultiReQA					
		NQ	SQuAD	HQA	SQA	TQA	Avg.
<i>Baselines</i>							
	BCM	46.38	86.33	77.49	70.41	61.35	68.39
	RM	47.15	86.57	74.71	70.15	61.34	67.98
	TM	51.64	86.67	68.57	69.37	63.64	67.98
<i>SCONER (Ours)</i>							
P@1	Q	50.44	90.06	71.54	71.26	65.22	69.70
	Q+A	50.54	89.97	77.63	77.74	66.52	72.48
	Q+KA	51.72	89.34	74.51	77.75	66.97	72.06
	KQ+KA	53.44	89.43	77.25	75.92	66.07	72.42
<i>Baselines</i>							
	BCM	58.02	91.30	84.98	78.70	71.11	76.82
	RM	58.46	91.35	83.16	78.47	71.07	76.50
	TM	61.57	91.10	79.80	79.33	73.99	77.16
<i>SCONER (Ours)</i>							
MRR	Q	59.71	92.96	77.83	78.49	72.13	76.22
	Q+A	60.42	92.92	84.33	84.14	74.06	79.17
	Q+KA	61.21	92.45	82.00	84.38	74.09	78.83
	KQ+KA	62.31	92.62	83.77	82.74	73.25	78.94

Table 4.3: We initialize each model using the STS model. We use green color to indicate increase compared to the corresponding result of vanilla RoBERTa, and red for a decrease. In most cases, the STS model is better than RoBERTa.

Label	Model				
	Q	Q+A	Q+KA	KQ+KA	RM
fix	92.51	92.36	92.71	91.92	90.58
generated	73.42↓	91.57↓	91.78↓	90.92↓	-

Table 4.4: Each model is initialized with RoBERTa model. Three SCONER using generated scores beat best baseline. Using score 5 is better than generated scores. ↓ means decrease compared to fix score.

#	Model MRR			BM25 Recall		
	SQuAD	NQ	HQA	SQuAD	NQ	HQA
50	91.77	60.50	86.30	94.85	73.65	99.29
100	91.77	61.66	86.21	96.55	77.19	99.29
150	92.54	61.55	86.03	97.56	79.09	99.68
200	92.71	61.50	85.88	98.04	80.20	99.80

Table 4.5: # means the number of re-ranking candidates and HQA means HotpotQA dataset. When the recall of a small size candidate is high (e.g. 99%), using small-size of candidates in re-ranking is better.

Chapter 5

MULTIMODAL RETRIEVER FOR KNOWLEDGE BASED VISUAL QUESTION ANSWERING

Initial work in IR dealt mainly with unimodal information retrieval, for e.g. using text queries to retrieve information. However, in some cases, a single modality (i.e. format) such as text may not be adequate for a query to convey all relevant cues. For instance, if someone spots a flower and wants to find where s/he can buy it, s/he would need to know the name of the flower – without knowing the name of the flower, it would be difficult to use a text query to find relevant information. However, multi-modal queries that combine an image of the flower taken with a mobile camera and the text phrase “shops that sell this flower” allow a user to convey such information. Multi-modal information retrieval (MMIR) seeks to provide algorithmic solutions to such problems where the query is composed of different formats and modalities.

In the last couple of years, highly impact technological advances have been made in the field of multi-modal representation learning, and using these learned representations have been used to improve zero-shot image classification (CLIP Radford *et al.* (2021a)), text-to-image synthesis (DALL-E Ramesh *et al.* (2021)), image captioning (OSCAR Li *et al.* (2020b)), and prompt-based vision-and-language tasks (FLAMINGO Alayrac *et al.* (2022)). These advances in shared image-text representations are also being used for information retrieval.

In the following, we will first describe a VQA benchmark that requires external knowledge, OkVQA (Marino *et al.*, 2019a). Then, we describe a knowledge collection process and multimodal retrievers that find knowledge from our collected corpus.

Lastly, the experiment results and analysis are presented.

5.1 An Overview of Knowledge-Based Question Answering

Knowledge-based VQA is a challenging task, where knowledge present in an image is not sufficient to answer a question. It requires a method to seek external knowledge. Figure 5.1 shows two examples from the OkVQA benchmark (Marino *et al.*, 2019a), which is normally used to study knowledge-based VQA. In each of the two examples, external knowledge is needed to answer the question. For instance, in the first example, to identify the vehicle used in the item shown in the image (top-left), a system needs to first ground the referred item as a fire hydrant and then seek external knowledge presented top-right of the image. The challenge is to ground the referred object in the image and retrieve relevant knowledge where the answer is present.

Although the OkVQA benchmark encourages a VQA system to rely on external resources to answer the question, it does not provide a knowledge corpus for a QA system to use. As such, existing methods rely on different resources such as ConceptNet Speer *et al.* (2017), WordNet Miller (1992), and Wikidata Vrandečić and Krötzsch (2014), resulting in the following issues:

1. It is difficult to fairly compare different VQA systems as it is unclear whether the difference in performance arises from differing model architectures or the different knowledge sources.
2. The different formats of the knowledge sources, such as the structured ConceptNet and the unstructured Wikipedia, demand different modules to retrieve knowledge, consequently making a knowledge-based VQA system complicated.
3. External resources like ConceptNet and WordNet have limitations. First, they only cover a limited amount of knowledge. For example, ConceptNet provides only 34 relation types, and there is a vast amount of knowledge that is hard to be described

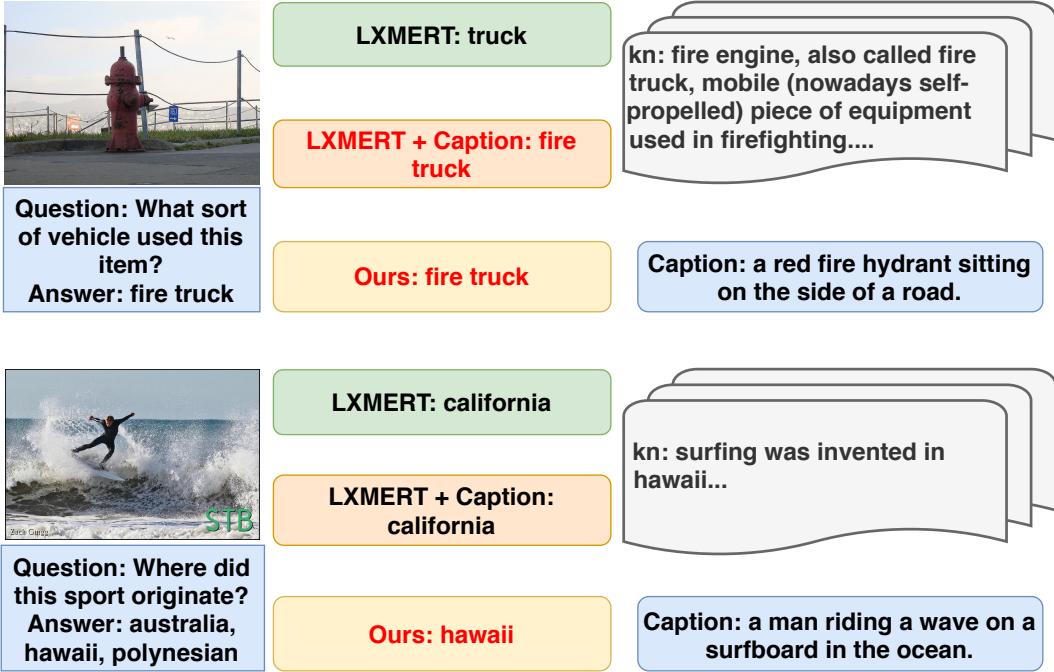


Figure 5.1: Two examples from OkVQA: the middle column are predictions by two baselines and one by our proposed Visual-Retriever-Reader pipeline. The left column is relevant knowledge and the corresponding captioning of images.

by a relation in a knowledge graph, such as *describe the logo of Apple Inc.* Second, constructing a structured knowledge base requires heavy human annotation and is not available in every domain. Thus, it limits the application of a knowledge-based VQA system that relies on a structured knowledge base.

5.2 A Knowledge Collection Approach

Motivated by the issue mentioned in the previous section, we collect a knowledge corpus for the OkVQA benchmark. Our corpus is automatically collected via Google Search¹ by using the training-split question and the corresponding answers, and we provide a training corpus with 112,724 knowledge sentences and a full testing corpus

¹<https://developers.google.com/custom-search/v1/>

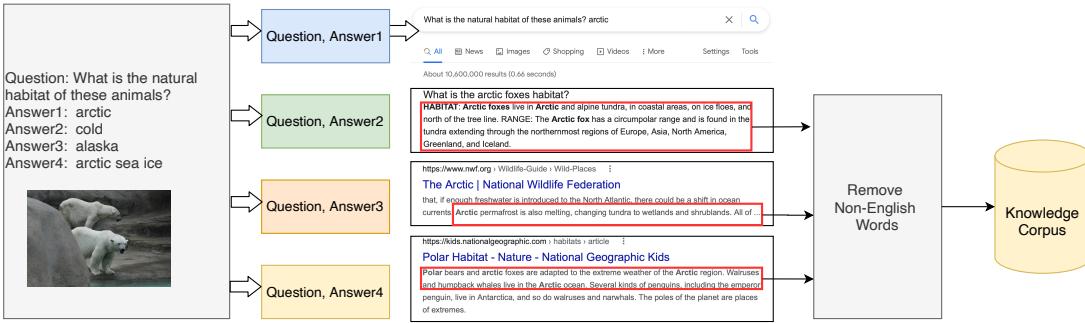


Figure 5.2: The overall process of Knowledge Corpus Creation. The question first combines the answers one by one to form a query, and then the query is sent to the Google Search API to retrieve the top 10 webpages. The knowledge is obtained from the snippet with further processing. Finally, we integrate the knowledge into the corpus. As shown on the search result page, the black boxes represent webpages, and the red boxes represent snippets.

with 168,306 knowledge sentences. The knowledge corpus is in a uniform format, i.e., natural language. Thus, it is easy to use other OkVQA methods. As we will show in the experiments section (§5.4), the knowledge base provides rich information to answer OkVQA questions. The overall process of knowledge corpus creation (Figure 5.2) consists of the following four steps.

Step 1: Query Preparation Based on the assumption that the knowledge used for answering training set questions can also help in testing, the OkVQA training questions are used with their answers to collect related knowledge from a search engine. We concatenate each question with each answer to get a “Question, Answer” pair. For example, in Figure 5.2, the question ”What is the natural habitat of these animals?” has four answers, and each answer is attached to the question one by one to construct four queries.

Step 2: Google Search Webpage The generated queries are sent to Google Search API to obtain knowledge. As presented in Figure 5.2, a good search result web page contains a title, a link, and a snippet that consists of multiple complete or

incomplete sentences and shows the most relevant part to the query. The top *ten* web pages with their snippets as the raw knowledge are chosen.

Step 3: Snippet Processing The snippets from Google search results consist of multiple sentences, some are complete but some are not. One option is to split snippets into multiple sentences, but the experimental result shows that sentence-level knowledge is worse than snippet-level. Thus, we choose to use snippets as knowledge. To address the incomplete sentence issue, we find and grab the complete sentence present on the webpage. After this pre-processing, ten snippet-knowledge from each “Question, Answer” query is selected.

Step 4: Knowledge Processing We first remove the duplicated data among each ”Question, Answer” pair. The long knowledge (more than 300 words) or short knowledge (less than ten words) are removed. Pycl2² is applied in this step to detect and remove the non-English part of each knowledge. Each knowledge is assigned a unique ID and duplicate knowledge sentences are removed. We curate in total 112,724 knowledge sentences for the OkVQA training set.

5.3 Multi-modal Retriever

We introduce two styles of visual retrievers: term-based and neural-network-based. In the neural style, we further introduce two variants.

5.3.1 Term-based Retriever.

In BM25 (Robertson and Zaragoza, 2009a), each query and document is represented by sparse vectors in d dimension space, where d is the vocabulary size. Then the score of a query and a document is computed based on the inverse term’s frequency.

²<https://pypi.org/project/pycl2/>

BM25 can only retrieve documents for a query in text format, but an image is part of a query in our task. To tackle this issue, we first generate image captions using a caption generation model. Then we concatenate the question and the caption as a query and obtain a list of documents by BM25.

5.3.2 Multimodal Neural Retriever.

Unlike BM25, neural retrievers extract the dense representations for a query and a context from the neural model(s). We use DPR (Karpukhin *et al.*, 2020a) as a neural retriever, which employs two BERT (Devlin *et al.*, 2019a) models to encode the query and context respectfully, then applies inner-dot product to estimate the relevancy between a query and a context. Similar to BM25, the DPR model considers the query in text format. To adapt DPR in the visual domain, we propose two methods. *Image-DPR*: we use LXMERT (Tan and Bansal, 2019a) as the question encoder, which takes the image and question as input and outputs a cross-modal representation. *Caption-DPR*: similar to the strategy we use in term-based retriever, we concatenate the question with the caption of an image as a query and use standard BERT as a query encoder to get the representation. In both *Image-DPR* and *Caption-DPR*, we use standard BERT as a context encoder. Figure 5.3 shows the architectures of standard DPR, *Image-DPR* and *Caption-DPR*. To train a neural retriever, we use the inner-dot product function to get the similarity score of relevant and irrelevant knowledge to a question and optimize the negative log-likelihood of the relevant knowledge.

Weak Supervision Training The retriever is trained using weak supervision, as the ground-truth knowledge context is unknown for a given question-image pair. Particularly, given a query and an image, we assume that knowledge that contains any of the answers is relevant, and we use the in-batch negative samples (Karpukhin

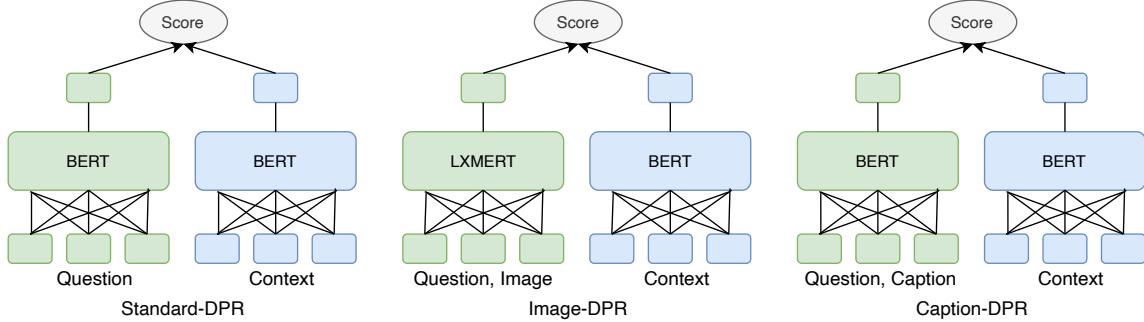


Figure 5.3: Comparison between standard DPR, Image-DPR, and Caption-DPR: while the context encoder is the same for three models, in standard BERT(left), the question encoder only takes a question as input, the Image-DPR(middle) takes both question and image as input, the Caption-DPR (right) takes the question and the caption as input.

et al., 2020a) for training, i.e., in the training time, any relevant knowledge for other questions in the same batch are considered as irrelevant.

5.4 Experiments and Results

We evaluate the performance of a retriever based on Precision and Recall. The two metrics are based on the assumption that any retrieved knowledge that contains any of the answers annotated in the OkVQA dataset is relevant. This assumption is because it is unknown which knowledge is relevant to a question-image pair. Therefore, the computation of Precision and Recall in our case is different from the traditional definition and illustrated as follows:

Precision Precision reveals the proportion of retrieved knowledge that contains any of the answers to a question-image pair. Mean Precision is the mean Precision of all question-image pairs. Mathematically,

$$P(Q, A, KN) = \frac{1}{K} \sum_{i=1}^{i=K} \min\left(\sum_{\substack{j=1 \\ A_j \in KN_i}}^{j=M} 1, 1 \right),$$

where Q is a question, KN is a list of retrieved knowledge, A is a list of correct answers, K is the number of KN , M is the number of A .

Recall Recall reveals if at least one knowledge sentence in the retrieved Knowledge contains any answers to a question-image pair. Mean Recall is the mean of the Recall of all question-image pairs. Mathematically,

$$R(Q, A, KN) = \min\left(\sum_{i=1}^{i=K} \sum_{\substack{j=1 \\ A_j \in KN_i}}^{j=M} 1, 1\right),$$

where the meaning of the symbols is the same as described in Precision.

Main Result. We evaluate retrievers' performance based on Precision and Recall. Table 5.1 shows that Caption-DPR consistently outperforms BM25 and Caption-DPR on the various number of retrieved knowledge. It is interesting to see that Caption-DPR outperforms BM25 significantly since BM25 is a hard-to-beat baseline in open-domain QA (Lee *et al.*, 2019b; Lewis *et al.*, 2020b; Ma *et al.*, 2021b).

Model	# of Retrieved Knowledge													
	1		5		10		20		50		80		100	
	P*	R*	P*	R*	P*	R*	P*	R*	P*	R*	P*	R*	P*	R*
BM25	37.63	37.63	35.21	56.72	34.03	67.02	32.62	75.90	29.99	84.56	28.46	88.21	27.69	89.91
Image-DPR	33.04	33.04	31.80	62.52	31.09	73.96	30.25	83.04	28.55	90.84	27.40	93.80	26.75	94.67
Caption-DPR	41.62	41.62	39.42	71.52	37.94	81.51	36.10	88.57	32.94	94.13	31.05	96.20	30.01	96.95

Table 5.1: Evaluation of three proposed visual retrievers on Precision and Recall: Caption-DPR achieves the highest Precision and Recall on all number of retrieved knowledge.

Effects of Completeness of Corpus. So far, when we test the model performance, we use the knowledge corpus collected only by training questions. However, if the entire training corpus does not include relevant knowledge to testing questions, our

Model	# of Retrieved Knowledge						
	1	5	10	20	50	80	100
BM25	+6.00	+6.28	+4.88	+4.32	+3.83	+3.17	+2.56
Image-DPR	+2.24	+2.60	+2.93	+2.29	+1.83	+1.29	+1.25
Caption-DPR	+8.88	+8.88	+7.04	+4.65	+2.98	+2.23	+1.88

Table 5.2: Recall increases when the Caption-DPR method retrieves knowledge from a complete knowledge corpus created using train and test questions.

model is under-evaluated because of the incompleteness of the knowledge corpus. To fairly see how our model performs when the knowledge corpus is complete, we use the same knowledge collection method described in §5.2 to collect knowledge for testing questions. Then we combine the training and testing knowledge as a complete corpus, which increases the corpus size from 112,724 to 168,306. We test how our multimodal retrievers perform on the complete corpus and the results are presented in Table 5.2. As we expected, a complete corpus is helpful for all three retrievers even though the corpus size increased, which demonstrates that our models do not overfit the training data and have great potential to be applied to real-life applications.

5.5 Discussion and Summary

To approach OkVQA challenge, we first collect an easy-to-use free-form natural language knowledge corpus for VQA tasks with external knowledge. Then we propose three multimodal retrievers that take a multimodal query as input and find the most relevant text knowledge to answer a such complex question. Our experiments show that among the three visual retrievers, the Caption-DPR performs the best and we also show the importance of the completeness of the knowledge and our models can generalize to a larger corpus. While the Caption-DPR performs the best, it relies

on the captioning generator, which might not always be available in some domains, thus, in the next chapter, we propose an end-to-end retriever that does not require the intermediate modules. Further, in the future chapter, we show that the visual retriever can improve the downstream task OkVQA.

Chapter 6

END-TO-END MULTIMODAL RETRIEVER: REVIZ

Humans retrieve information using many hints and cues – for instance if we forget the name “leopard” but want to explain the concept to someone else, we could show a picture of a tiger and say “it is an animal that looks like this, but has spots instead of stripes”. When children learn to draw a new shape like an *oval*, teachers often prompt them by showing a circle, but saying “make the circle stretched-out”. This method of learning new concepts from visual aids and language descriptions is a common way of reinforcing existing knowledge and allowing learners to explore and retrieve new concepts. We propose a task for vision-language models to retrieve knowledge with multi-modal queries, i.e. queries in which hints about the information to be retrieved are split across image and text inputs. Figure 6.1 contains an example of this task, where the image shows the Empire State Building in New York City. If we retrieve knowledge using only the image, is it likely that the retrieved information (K1) will be related to the Empire State Building. However, K1 is insufficient to answer the question. On the other hand, if we retrieve knowledge using only the question, then the information retrieved (K2) is likely to be related to the tallest building in all cities (and not restricted to New York City). K2 by itself is also insufficient to answer the question. This example shows that the combined query containing both image and text (question) is necessary for retrieving relevant knowledge (K3).

We introduce a new benchmark and dataset called ReMuQ (**R**etrieval with **M**ultimodal **Q**ueries) to train and evaluate models to retrieve the answer from a corpus given multimodal (vision + language) queries. To create multimodal queries, we start with the WebQA (Chang *et al.*, 2021) dataset as a source – WebQA contains



Question: *Is this the tallest building in the city?*
External Knowledge

K1: The Empire State Building is a 102-story Art Deco skyscraper in Midtown Manhattan, New York City.

K2: The 828-metre (2,717 ft) tall Burj Khalifa in Dubai has been the tallest building since 2010. The Burj Khalifa has been classified as megatall.

K3: The tallest building in New York is One World Trade Center, which rises 1,776 feet (541 m).

Figure 6.1: The image shows the Empire State Building, and the question asks if it is the tallest building in “the city” (New York). K1 is retrieved by using only the image, K2 is retrieved by only using the question, and K3 is retrieved using both image and the question. Only K3 can be used to answer the question correctly.

images annotated with questions and answers. We select questions from WebQA where the answer includes both an image and text. We then remove any image information from text and combine the image and the augmented text to form a new multimodal query. We also construct a large retrieval corpus as the source of knowledge for this task.

This task requires integrating the contents from both modalities and retrieve knowledge – in this paper we denote such a system as a “VL-Retriever”. Existing VL-Retrievers (Qu *et al.*, 2021a; Luo *et al.*, 2021; Gao *et al.*, 2022) typically follow a two-step process to retrieve knowledge: (1) converting the image into captions or keywords, appending them to the text query, and (2) using a text-retriever system to retrieve the knowledge. However, this approach can result in a loss of important information from the image, such as context and background. Additionally, using a caption generation model trained on a particular domain does not transfer well to other domains in real-world applications.

To address these issues, we propose an end-to-end VL-Retriever that has the

potential to leverage the entire image, rather than just object categories, keywords, and captions. We call this model *ReViz*, a retriever model for “**R**eading and **V**izualizing” the query. As part of ReViz, we use a vision transformer-based model, ViLT (Kim *et al.*, 2021), to directly encode the image from raw pixels with context inputs, and we employ BERT (Devlin *et al.*, 2018) as the knowledge encoder to represent the long, free-form text as a knowledge embedding. ReViz differs from previous retrieval models in two main ways. First, it does not require an extra cross-modal translator (e.g., a captioning model) or object detector to represent the images. Second, its end-to-end design allows for the flexible retraining of each submodule of the model, which can mitigate potential issues caused by domain gaps.

Unlike neural text-retrievers (Karpukhin *et al.*, 2020a), the query and knowledge encoders in ReViz are of different types of modality (i.e. multimodal transformer and language transformer). The different semantic spaces of the query and knowledge embeddings make alignment between them difficult. To address this, we propose a novel multimodal retrieval pretraining task. To create training data, we construct triplets of (input-image, input-text, output-knowledge) from WiT (Srinivasan *et al.*, 2021), a large dataset of encyclopedia-type knowledge collected from Wikipedia. We process the data such that the input image and text have mutually exclusive information.

Our contributions and findings are listed below.

- We introduce a new dataset *ReMuQ* to facilitate research on retrieval with multi-modal queries.
- We propose an end-to-end VL-Retriever, ReViz, that directly acquires knowledge given multimodal query. ReViz is not dependent on any cross-modal translator, such as an image captioning model or an object detector.
- We pretrain ReViz on a novel multimodal retrieval pretraining task, VL-ICT. Empirically, we observe that with the proposed pre-training on the WiT dataset, our

VL-Retriever is a powerful zero-shot multimodal retriever that surpasses existing single-modal knowledge retrieval methods.

6.1 Related Work

Cross-Modal Retrieval aims to find information from a different modality than the query; for instance retrieving images from text (text-to-image), text from images (image-to-text) Young *et al.* (2014); Chen *et al.* (2015), text-to-video and video-to-text Rohrbach *et al.* (2015); Xu *et al.* (2016); Zhou *et al.* (2018). In contrast, we consider retrieval of knowledge for queries comprised of both modalities (i.e. image and text) together.

Knowledge-based question answering. Retrievers are important for finding relevant knowledge to aid knowledge-based question-answering models for tasks such as FVQA Wang *et al.* (2017) (commonsense knowledge), Text-KVQA Singh *et al.* (2019) which requires knowledge of the text in the image, and KVQA Shah *et al.* (2019)(world knowledge about named entities). Both FVQA and KVQA are equipped with knowledge graph as external corpus. In OKVQA Marino *et al.* (2019b) and its augmented versions S3VQA Jain *et al.* (2021) and A-OKVQA Schwenk *et al.* (2022), models are free to use any existing knowledge bases to retrieve relevant knowledge. WebQA Chang *et al.* (2021) is a multi-hop reasoning dataset that requires a system to aggregate multiple sources to answer a question, where the answers can be found either via image search or general web search.

Knowledge-Retrieval with Multimodal Queries. While there are methods for retrieving knowledge from knowledge graphs (Narasimhan *et al.*, 2018; Li *et al.*, 2020a; Marino *et al.*, 2021), in this work, we focus on systems that retrieve knowledge from

free-form text, which is more readily available and comprehensive. Previous methods involve converting images into language representations such as captions Qu *et al.* (2021a); Gao *et al.* (2022) or object tags Gui *et al.* (2021); Yang *et al.* (2021), and then using a text-based retriever such as BM25 Robertson and Zaragoza (2009b) or DPR Karpukhin *et al.* (2020a) to find relevant knowledge. Gao *et al.* (2022) leverage GPT-3 Brown *et al.* (2020) to generate the knowledge. Qu *et al.* (2021a); Luo *et al.* (2021) use a vision and language model to obtain cross-modal representations. CLIP Radford *et al.* (2021b) has also been applied to retrieval tasks; however it has limitations due to its separate encoding of text and image without a multi-modal fusion module.

6.2 Retrieval with Multimodal Queries

In this section, we define the problem statement for knowledge retrieval with multimodal queries and describe the construction of the ReMuQ dataset to assess models performing this task.

6.2.1 Problem Statement

Given a query $Q = (I, T)$ containing as image I and text T , we wish to learn a mapping to relevant textual knowledge K from a corpus C . Note that the two modalities I and T are such that each contains partial information about K . Both I and T are necessary for successful retrieval of K and Only using one of the two modalities is inadequate.

6.2.2 ReMuQ Dataset Creation

In ReMuQ each query has exactly one ground truth knowledge associated with it. To create such queries, we augment WebQA questions Chang *et al.* (2021), and collect

a large corpus to serve as the knowledge source for any retrieval systems. WebQA is a multihop and multimodalQA dataset including text questions of different types such as Yes/No, Open-ended (e.g. shape, color, etc.), and multi choice (MC) questions. The images are crawled from Wikimedia Commons, both questions and text answers are created by annotators.

To create multimodal queries, we utilize the MC questions in WebQA, which are associated with multiple choices as knowledge sources in the form of text or images. The ground truth answers of the questions include text-only, image-only, or both text and image. We adapt important steps to create multimodal queries and explain the pipeline of the curation procedure below and in Figure 6.2.

- (1) **Question Filtering.** We select multiple-choice questions which have answer choices containing both image and text.
- (2) **Multimodal Query Construction.** The initial multimodal query is the combination of the question and the corresponding image. In order to enforce a system to integrate information from both text and images, we use $tf-idf$ to select keywords and then remove them in the question. Our new multimodal-query is then the concatenation of the augmented question and the image, with the text-answer to be the ground-truth knowledge.
- (3) **Retrieval Corpus Construction.** We aggregate the textual knowledge from all samples as the common knowledge corpus for multimodal retrieval, resulting in a large corpus of $\sim 199k$ knowledge descriptions.
- (4) **Dataset Train-Test Split.** We divide ReMuQ into 70% for training and 30% as testing split. The new curated dataset contains 8418 training samples and 3609 testing samples, together with a knowledge corpus with 195,837 knowledge descriptions.

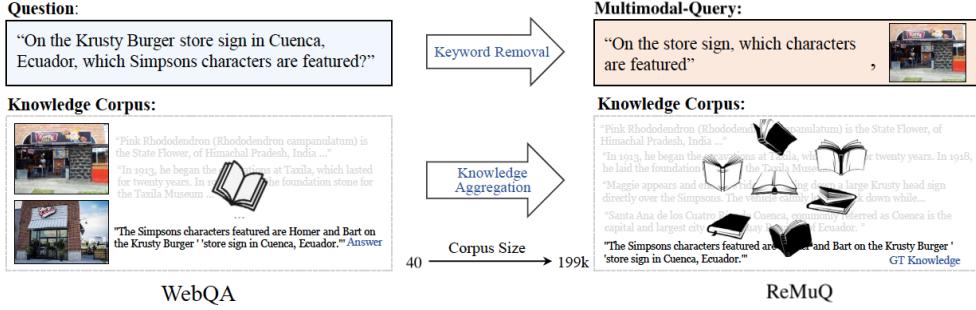


Figure 6.2: Dataset creation Procedure for ReMuQ. We use WebQA as source of the raw data. The multimodal-Query in ReMuQ is the combination of an image and the question from WebQA where the overlapped information with the image is removed. The ground truth knowledge of ReMuQ is the answer from WebQA. The corpus consists of all answers and the distracted knowledge candidates given in ReMuQ.

6.3 ReViz Model

Prior work on Vision-Language (VL)-Retrievers has focused on two-stage methods where the first stage involves feature-extraction using pretrained visual and textual encoders and the second stage learns retrieval using these features. A typical VL-Retriever can be expressed as:

$$K = \text{VL-RETRIEVER}(T, F; C), \quad (6.1)$$

where C is the knowledge corpus, T is the text component of the query, and F denotes the extracted features of image I . This feature extraction can be done in two ways; (1) by converting the visual inputs into a human-readable textual description via an image captioning model or a series of object tags by object detector, (2) by extracting object features using an object detector.

End-to-End VL-Retriever. Instead, in this work, we are interested in building an end-to-end VL-Retriever, that encodes and selects the knowledge from the corpus using a VL model:

$$K = \text{VL-RETRIEVER}(T, I, C). \quad (6.2)$$

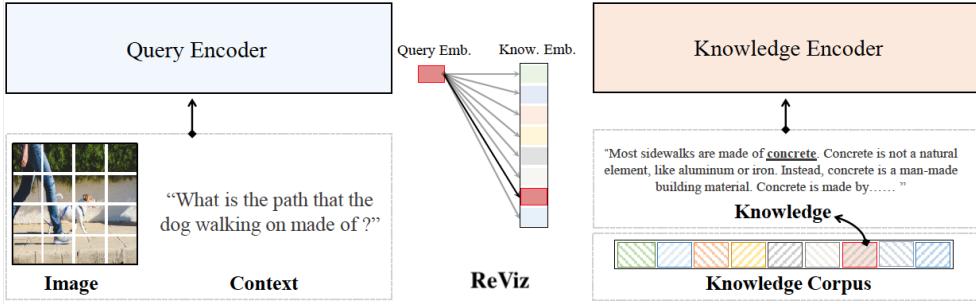


Figure 6.3: Overall architecture of our proposed ReViz. Our model consists of a Vision-Language Transformer that encodes the image and text, meanwhile the knowledge encoder projects the knowledge into knowledge embedding. During inference, our model selects the knowledge from the corpus that has the largest relevance score with the image-text embedding.

We propose ReViz, an end-to-end VL-RETRIEVER that learns to maximize the multimodal query and knowledge similarity for knowledge retrieval tasks. We introduce its architecture below.

6.3.1 Model Architecture

ReViz can read and visualize the input query, consists of two components, the multimodal query encoder and the knowledge encoder. Figure 6.3 illustrates the pipeline of our model.

Multimodal Query Encoder. We use ViLT Kim *et al.* (2021) to jointly encode the text input T and the image I . In ViLT, an image is first partitioned into a set of a fixed size of patches – these patches are encoded as continuous visual tokens through a linear projection layer Dosovitskiy *et al.* (2020). These visual tokens are concatenated with the text tokens and summed with the position embeddings and fed into a stack of several self-attention blocks. The final multimodal representation is obtained by applying linear projection and hyperbolic tangent upon the first index token embedding.

$$\mathbf{Z}_q = \text{ViLT}(I, T) \quad (6.3)$$

Knowledge Encoder. To encode knowledge, we use a pre-trained BERT (Devlin *et al.*, 2018) model, which produces a list of dense vectors (h_1, \dots, h_n) for each input token, and the final representation is the vector representation of special token [CLS].

$$\mathbf{Z}_k = \text{BERT}(K) \quad (6.4)$$

After the embeddings of query and knowledge are computed by the encoders, inner-dot product of the embeddings is considered as the relevancy score.

$$\text{Score}(I, T, K) = \mathbf{Z}_k^\top \cdot \mathbf{Z}_q \quad (6.5)$$

6.3.2 Training

The training objective of ReViz draws inspiration from the instance discrimination principle based on contrastive learning. The loss function to be minimized is given below:

$$\mathcal{L} = -\log \frac{\exp(\mathbf{z}_q \cdot \mathbf{z}_k)}{\exp(\mathbf{z}_q \cdot \mathbf{z}_k) + \sum_{\hat{k} \in \mathbf{B}_k, \hat{k} \neq k} \exp(\mathbf{z}_q \cdot \mathbf{z}_{\hat{k}})}, \quad (6.6)$$

where \mathbf{z}_q denotes the query embedding, \mathbf{z}_k denotes the relevant knowledge embedding, and $\mathbf{z}_{\hat{k}}$ is the irrelevant knowledge embedding which serves as negative instances. We use all in-batch samples (\mathbf{B}_k) as the negative instances.

Training with Hard Negative. Adopting random samples as negative instances may cause sub-optimal metric space. Existing work shows that mining with hard negative samples leads to discriminative representations and has been applied to a broad series of tasks like face recognition Zhang *et al.* (2017), object detector Shrivastava *et al.* (2016), and metric learning for retrieval tasks Faghri *et al.* (2017); Harwood *et al.* (2017). Inspired by this, we also experiment with the hard negative technique to further

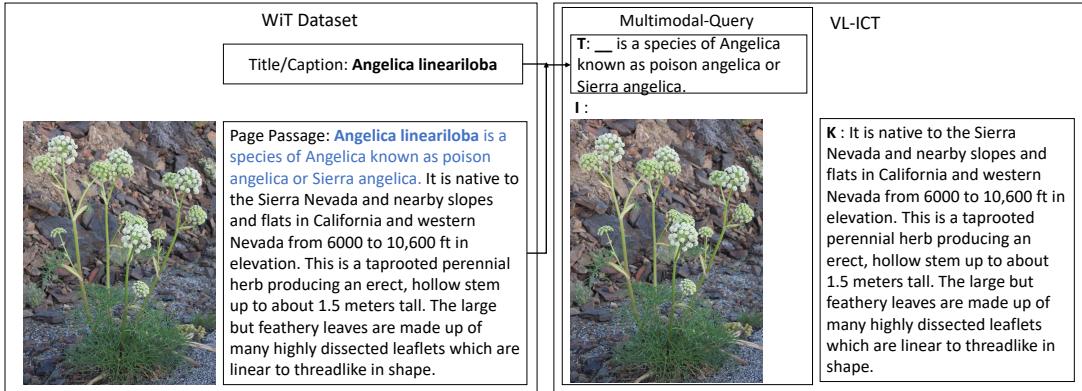


Figure 6.4: Figure on the left shows an example of the WIT dataset, crawled from Wikipedia. Figure on the right shows our constructed (T, I, K) triplet: T is a sentence from the passage and the words overlapped with the title/caption is masked; K is the remaining passage after removing the sentence.

boost the retrieval performance. To obtain the meaningful hard negative samples, we first train ReViz with the supervisions in *eq. 6.6*. With that, for each training question, we retrieve the top-100 knowledge instances (excluding the ground-truth) as the hard negative samples. Note that we only apply hard negative mining to fine-tuning on downstream task but not the pretraining task (introduced in the next section).

6.4 Pretraining Task for VL Retriever

Previous work (Chang *et al.*, 2020; Lee *et al.*, 2019b; Guu *et al.*, 2020a) suggests that pretraining a retriever on unsupervised task that closely resembles retrieval can greatly improve the downstream tasks performance. We propose a pretraining task called VL-ICT, which is inspired by ICT Lee *et al.* (2019b) task in NLP domain.

ICT aims to train text-based information retrieval (IR) system for the open-domain question answering task. To train a model without annotated data, Lee *et al.* (2019b) propose to construct pseudo $(question, context)$ pairs as the training data for IR system. In particular, given a passage P , a random sentence S in the passage is selected as the pseudo question, and the remaining passage P' is considered as the

relevant context. Such a weakly-supervised setting enables large-scale ICT pre-training, leveraging any available knowledge base as the training corpus.

VL-ICT. We propose VL-ICT task to pre-train ReViz, which can be applied to multi-modal scenarios when both language and vision inputs exist in the query. In VL-ICT, a (I, T, K) triplet is used for training. Importantly, I and T , contain mutually exclusive information and are both necessary for knowledge retrieval. However, such condition is not naturally existing, thus, we propose an automatic procedure to construct triplet satisfying this condition in the following.

VL-ICT Training Data. Figure 6.4 shows a snapshot of our data construction process where we use the WiT dataset Srinivasan *et al.* (2021) as the source. Each WiT entry provides a title of the page or an image caption, a passage, and an image. We use the image from this WiT entry as the image I in our VL-ICT triplet. We observe that the title or caption is usually entities, it allows us to simply use word matching to find the sentences in the page passage that include the title/caption. We take such sentences as the text (T), then we remove this sentence from the passage and use the remaining passage as the knowledge (K). To enforce that (T) and (I) have mutually exclusive but important information, we mask keywords in T that appear in both T as well as the caption. In our experiments, we only select the English entities in WiT and execute the above process, and this results in 3.2 million (I, T, K) training triplets.

6.5 Experiments and Results

Datasets. In addition to ReMuQ, we conduct experiments on OKVQA to obtain stronger evidence for the efficacy of our method. Here, instead of QA task, we use

Model	Dataset	KB-Size	Metric						
			MRR@5	P@5	R@5	R@10	R@20	R@50	R@100
CLIP-IMG+Q	OKVQA	GS-112K	19.08	11.13	34.54	50.48	65.08	80.62	88.11
BM25 (GenCap)	OKVQA	GS-112K	36.36	27.54	51.35	63.04	73.37	84.21	90.39
DPR (GenCap)	OKVQA	GS-112K	39.15	27.72	55.56	66.44	75.59	87.17	92.42
ReViz+VL-ICT	OKVQA	GS-112K	45.77	33.18	64.05	75.39	84.21	91.64	94.59
TRiG Gao <i>et al.</i> (2022)	OKVQA	Wiki-21M	-	-	45.83	57.88	72.11	80.49	86.56
CLIP-IMG+Q	OKVQA	Wiki-21M	16.45	9.66	29.81	43.00	55.73	72.73	82.26
BM25 (GenCap)	OKVQA	Wiki-21M	36.43	27.89	50.16	60.92	71.62	82.82	88.74
DPR (GenCap)	OKVQA	Wiki-21M	41.15	28.10	59.41	71.13	81.73	89.90	93.39
ReViz+VL-ICT	OKVQA	Wiki-21M	44.03	32.94	62.43	73.44	82.28	89.93	93.76
CLIP-IMG+Q	ReMuQ	199K	0.34	0.17	0.78	1.36	2.41	7.34	47.88
BM25 (GenCap)	ReMuQ	199K	3.80	5.59	8.78	10.75	12.88	15.88	17.98
DPR (GenCap)	ReMuQ	199K	31.23	35.79	43.42	48.77	54.47	61.40	67.30
ReViz+VL-ICT	ReMuQ	199K	23.61	29.52	39.43	46.77	53.56	63.70	71.13

Table 6.1: Zero-shot performance of ReViz and baselines on two datasets: OKVQA and ReMuQ. OKVQA is evaluated on two knowledge sources. ReViz shows superior zero-shot performance in majority of the cases.

OKVQA as a testbed for retrieval task, i.e. to retrieve a relevant knowledge to a question such that it contains the answer span. Furthermore, we use two corpora, a small corpus collected from Google search API introduced in Luo *et al.* (2021), and a large corpus which contains 21M Wikipedia knowledge used in Gao *et al.* (2022).

Evaluation Metrics. Following Gao *et al.* (2022); Luo *et al.* (2021), we evaluate the performances of models by Precision@K (P@K), Recall@K (R@K), and MRR@5. We use similar metrics to evaluate the ReMuQ challenge except that P@1 is used instead of P@5 since ReMuQ has exactly one correct knowledge per query.

6.5.1 Zero-shot Retrieval

We first introduce three zero-shot baselines and then present the results.

Model	Dataset	KB-Size	Metric						
			MRR@5	P@5	R@5	R@10	R@20	R@50	R@100
ReViz	OKVQA	GS-112K	46.92	34.51	66.05	77.80	86.33	93.34	95.90
ReViz+VL-ICT	OKVQA	GS-112K	54.47	41.74	73.35	83.17	89.56	94.73	96.81
ReViz	OKVQA	Wiki-21M	41.66	30.08	60.88	72.20	81.07	89.16	93.10
ReViz+VL-ICT	OKVQA	Wiki-21M	43.68	31.36	61.91	72.63	81.05	89.28	93.44
ReViz	ReMuQ	199K	41.03	49.08	62.40	71.63	78.92	86.60	92.17
ReViz+VL-ICT	ReMuQ	199K	53.39	62.11	76.23	83.32	88.56	93.41	96.12

Table 6.2: Comparison of ReViz when it is fine-tuned on downstream tasks. We compare ReViz and ReViz+VL-ICT (our pretraining task). VL-ICT enables ReViz to be a stronger multimodal-query retrieval model.

CLIP Baseline. CLIP Radford *et al.* (2021b) is a vision-language model pre-trained on over 400M image-text pairs. We encode all knowledge descriptions via CLIP’s textual encoder **K**. Then, given an image-text pair as the query, we use the image encoder to get the visual representations (**I**) and use the textual encoder to get the embedding of **Q**. We compute the inner-dot products between all encoded visual representations (**I**) and **K** to get the top-100 knowledge for evaluation, similarly for **Q**. Finally we sum the scores and re-rank the top-100 knowledge. We find this performs the best than using individual modality (see Appendix).

BM25 Baseline. BM25 Robertson and Zaragoza (2009b) is a well-known efficient retrieval algorithm for text-based retrieval task based on the sparse representation. We use the caption of the image to represent the information of the image and thus we convert the multi-modal knowledge retrieval task into a pure text-based retrieval task.

DPR Baseline We adopt DPR Karpukhin *et al.* (2020a) trained on NaturalQuestions Kwiatkowski *et al.* (2019b) dataset as a baseline, to retrieve the knowledge given an input image-text pair. First, we use the contextual encoder of DPR to index the corpus, then we concatenate the question and the caption of the image as a

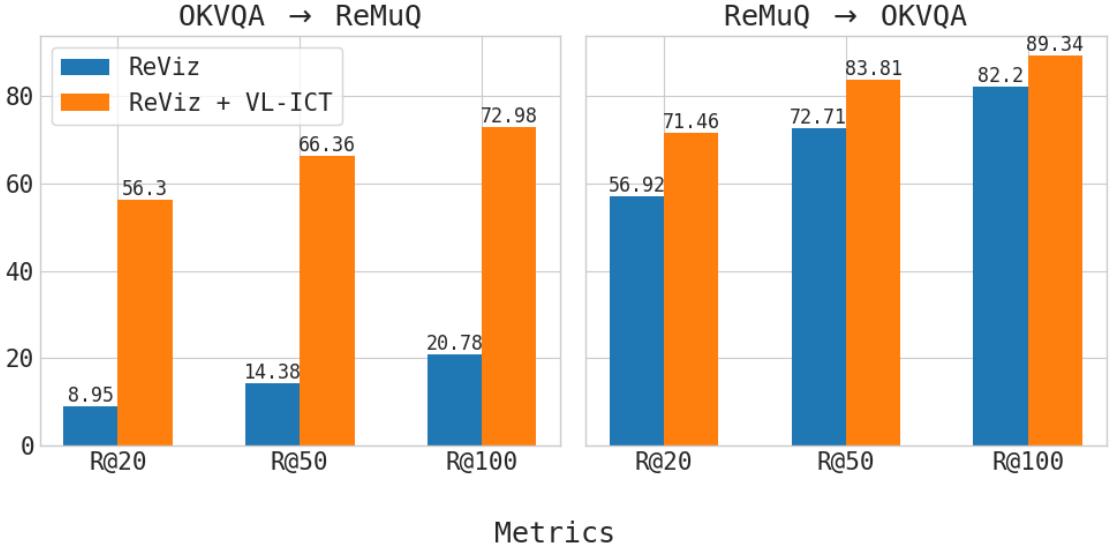


Figure 6.5: Evaluation of out-of-domain performances of ReViz and ReViz+VL-ICT. For OKVQA, we retrieve knowledge from GS-112K corpus. VL-ICT substantially improves the generalization of ReViz. X→Y denotes using X as the training domain and Y as the testing domain.

joint textual query. With that, the question encoder of the DPR extracts the dense representation of the query for later computation. Lastly, we retain the most relevant knowledge pieces by calculating the inner-dot product between the query and the knowledge embedding.

Result Table 6.1 shows the performances of baselines as well as ReViz pretrained on VL-ICT task. Among the baselines, we see that DPR is the strongest baselines. Surprisingly, although CLIP has shown strong performance on many classification and cross-modality pretraining task, it does not perform well on multimodal query retrieval task, this suggests that multimodal query retrieval is a challenging task for VL model. More importantly, we observe clearly that ReViz outperforms the baselines in terms of all metrics on OKVQA task on corpus of small and large size. On the ReMuQ dataset, ReViz wins CLIP and BM25 on all metrics, and DPR on two metrics. This demonstrates the effectiveness of our proposed pretraining task and the model

Model	FT	KB-Size	Metric						
			MRR@5	P@5	R@5	R@10	R@20	R@50	R@100
VRR-IMG Luo <i>et al.</i> (2021)	✓	GS-112K	-	31.80	62.52	73.96	83.04	90.84	94.67
VRR-CAP Luo <i>et al.</i> (2021)	✓	GS-112K	-	39.42	71.52	81.51	88.57	94.13	96.95
ReViz+VL-ICT	✓	GS-112K	54.47	41.74	73.35	83.17	89.56	94.73	96.81
TRiG Gao <i>et al.</i> (2022)	✗	Wiki-21M	-	-	45.83	57.88	72.11	80.49	86.56
ReViz+VL-ICT	✗	Wiki-21M	44.03	32.94	62.43	73.44	82.28	89.93	93.76

Table 6.3: Comparison of our best model with existing models on OKVQA. “FT” denotes fine-tuning. Our model surpasses existing methods by significant margins with or without fine-tuning and with different knowledge corpus.

design.

6.5.2 Fine-tuning Performance

To further demonstrate the effectiveness of VL-ICT pretraining task, we finetune models on downstream tasks and compare performance. We compare two versions of ReViz: (1) ReViz directly trained on the downstream task and (2) ReViz first pretrained on VL-ICT and then finetuned the down-stream task. In addition, We study two senarios: in-domain, where a model is trained on the training set of X domain and evaluated on the testing set of X; out-of-domain, where a model is trained on the training set of X domain and evaluated on the testing set of Y domain.

In-Domain Results. Table 6.2 shows the in-domain performance. On both datasets, pretrained ReViz consistently outperform vanilla ReViz, suggesting that the pretraining task equips ReViz better alignment between the multimodal queries and the relevant knowledge.

Out-of-Domain Results. We investigate if the VL-ICT pretraining task can improve the generalization of ReViz. We study the performances of ReViz under two

settings: train on OKVQA (domain **X**) and test on ReMuQ (domain **Y**); and the inverse. Table 6.5 shows that ReViz+VL-ICT+**X** shows obviously better results than ReViz+**X** on **Y**, especially when **X** is OKVQA and **Y** is ReMuQ. This suggests that models pre-trained with VL-ICT tasks are more robust than models without VL-ICT. We also see that the generalization performance still has a large gap with the fine-tuning, which suggests that OKVQA and ReMuQ are quite different tasks, and ReMuQ can be a good complement to OKVQA to study multimodal query retrieval task.

6.5.3 Compare ReViz with Existing Methods

We compare ReViz with existing retrieval methods for the OKVQA task. Note that most of the models on the leaderboard of OKVQA only report the final question answering accuracy but not the retrieval performance. In our experiments we include systems which report the retrieval performance.

Baselines. Luo *et al.* (2021) present two fine-tuned multimodal retrievers: VRR-IMG which uses LXMERT Tan and Bansal (2019b) and VRR-CAP to convert the image into captions for knowledge retrieval. Both retrievers use GS-112K as the knowledge corpus. TriG Gao *et al.* (2022) uses zeroshot retriever and Wikipedia 21M as the knowledge corpus. Since these systems use either fine-tuned retriever or zero-shot retrievers, for fair comparison, we compare the best fine-tuned model and zeroshot model with the corresponding corpus.

Results. In the fine-tuning scenario, in majority of the cases (only one exception, R@100), our models consistently shows better performance than previous methods overall metrics. Similarly, in the zero-shot case, our model is better than previous

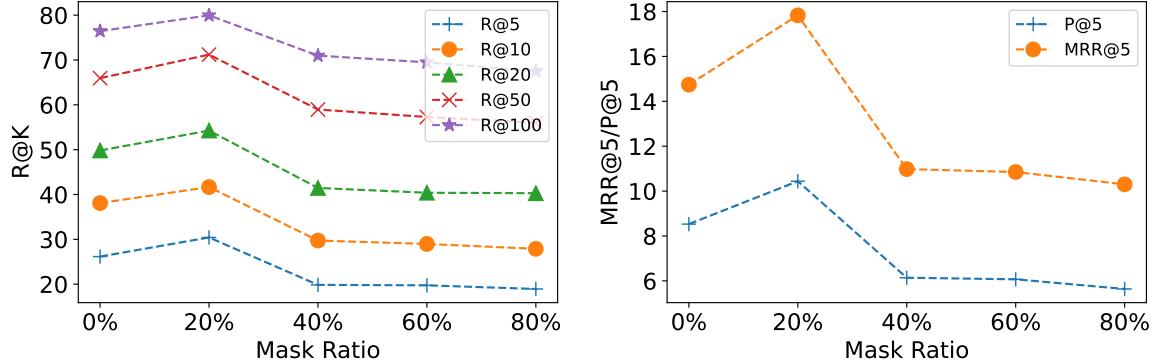


Figure 6.6: Effect of the masking ratio of sentences in VL-ICT task on ReViz’s performance on OKVQA Task. We use GS112K as the knowledge corpus.

model on all metrics by large margins.

6.5.4 Effects of Mask Ratio in VL-ICT Task

In VL-ICT, we mask the keywords in the sentence to prevent information leakage. Despite this, we find that the certain masked sentences still somehow overlap with the retrieved knowledge. We conjecture that this overlapping makes the VL-ICL task inevitably easy, and thus impairs the effects of pre-training. To study the optimal mask ratio, we conduct experiments to randomly mask the words in the sentence by different ratios. This study is performed on a smaller corpus of 1M VL-ICT training triplets and models are trained for one epoch. Figure 6.6 shows the results. We observe that removing 20% of the keywords yields the best performance amongst all ratios and is also better than maintaining the sentences intact (0% masking).

6.5.5 Effect of Generated Captions.

Previous systems which rely on the caption generation model are affected by the quality of generated captions. This may hamper the retrieval performance when the caption generation model is not trained on the same domain as the downstream task. In our ReMuQ dataset, the images are from Wikipedia, but the caption generator is

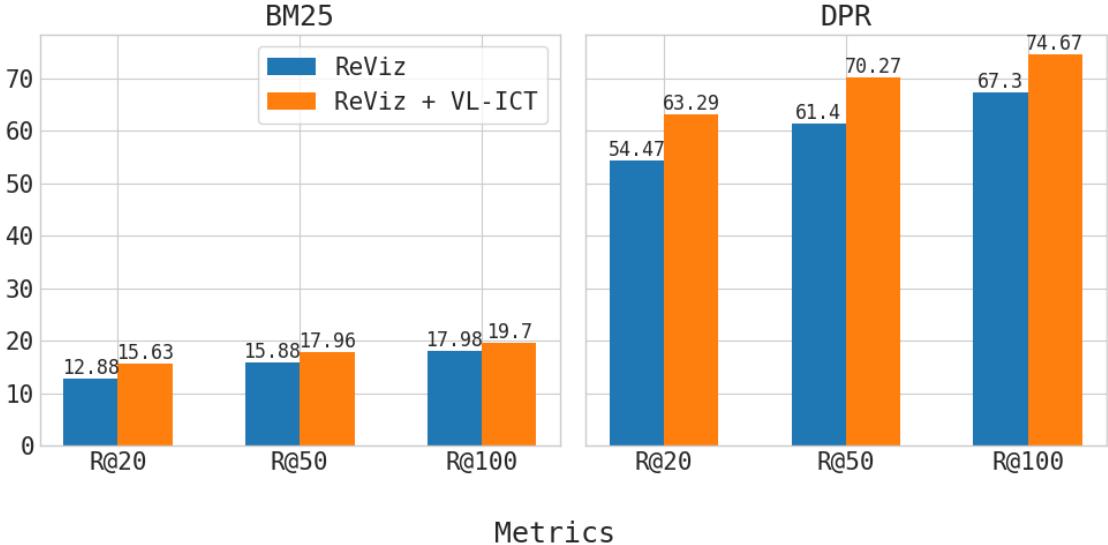


Figure 6.7: Comparison of captioning-dependent retrievers using generated captions and ground truth captions. The ground truth captions always lead to better performance than generated caption.

trained on MS-COCO (Lin *et al.*, 2014). We compare our two baselines, BM25 and DPR, using ground-truth image captions and the generated captions. Table 6.7 shows that using the ground truth caption is much better than the generated caption in all cases. This suggests that the caption generator is the bottleneck of the retrieval methods to convert the image information to image captioning. This demonstrates the limitations of previous methods and justifies our exploration of end-to-end training.

6.6 Discussion and Summary

During the creation of the ReMuQ dataset, we simply remove the words in the question that are duplicated in the image caption – in some cases, this may result in grammatical errors in the text query. We performed the experiments for studying optimal masking ratio on a subset of the pretraining data, due to resource constraints. We study knowledge retrieval with multi-modal (vision and language) queries, which, compared with existing retrieval tasks, is more challenging and under-

explored. We propose an end-to-end VL-retriever model, ReViz, which does not rely on any intermediate image to text translation modules. A novel weakly-supervised task (VL-ICT) is proposed to enable large-scale pre-training. With the existing benchmark OKVQA and our new curated challenging testbed (ReMuQ), our extensive evaluations show that our ReViz exhibits strong performance amongst all retrieval models in both zero-shot and fine-tuning scenarios.

Chapter 7

READING COMPREHENSION MODELS FOR TEXT

Machine reading comprehension (MRC) is to teach a system to understand the natural language and one of the long-standing challenge in Artificial Intelligent Chen (2018). Question Answering (QA) is an important task to evaluate the MRC capacity of an intelligent system and can be directly applied to real applications such as search engines (Kwiatkowski *et al.*, 2019a) and dialogue systems (Reddy *et al.*, 2019; Choi *et al.*, 2018). The state-of-the-art QA systems (termed as reader in this proposal) are mainly two types. Extractive readers (Seo *et al.*, 2017; Devlin *et al.*, 2019a) are widely used to tackle such a task, where the goal is to classify start and end positions of the answer in the context. Generative readers (Raffel *et al.*, 2020; Lewis *et al.*, 2020b; Izacard and Grave, 2021) have also shown remarkable performance, where the goal is to generate answers by autoregressively predicting tokens.

7.1 Extractive Reader

In extractive reader, an encoder firstly receives the concatenation of a question $\mathbf{q} : \{q_1, \dots, q_t\}$ and a context $\mathbf{c} : \{c_1, \dots, c_m\}$, where q_i and c_j are tokens in question and context, respectively. Then, it produces $\mathbf{h} : [h_1 | \dots | h_m] \in \mathbb{R}^{d \times m}$, where h_j corresponds to the d -dimensional contextual representation of context token c_j . We then stack two linear layers on top of the contextual representations to independently predict the probability of each context token being start and end positions of the correct answer. More formally, given a tuple $(\mathbf{q}, \mathbf{c}, \mathbf{a})$, where \mathbf{a} is an answer, the training objective is to minimize the following loss function

$$\mathcal{L}_{\text{Ext}} = -\log(\mathbf{P}_{\text{start,s}}) - \log(\mathbf{P}_{\text{end,e}}) \quad (7.1)$$

where $\mathbf{P}_{\text{start}}, \mathbf{P}_{\text{end}} \in \mathbb{R}^m$ are defined by

$$\mathbf{P}_{\text{start}} = \text{softmax}(\mathbf{w}_{\text{start}} \mathbf{h}) \quad (7.2)$$

$$\mathbf{P}_{\text{end}} = \text{softmax}(\mathbf{w}_{\text{end}} \mathbf{h}) \quad (7.3)$$

where $\mathbf{w}_{\text{start}}$ and \mathbf{w}_{end} denote for the linear layers to predict start and end tokens, $\mathbf{P}_{\text{start},s}$ and $\mathbf{P}_{\text{end},e}$ denote the probability of the ground truth start and end tokens of answer \mathbf{a} , respectively. In testing time, the answer span is decoded by $\text{argmax}_{i,j}\{\mathbf{P}_{\text{start},i} \times \mathbf{P}_{\text{end},j}\}$.

In this work, we have two variants of extractive readers. One is encoder-only models to get the contextual representation of each token. We call such kind of reader as **E-Extractive reader**. Apart from taking the conventional PrLMs such as RoBERTa and ELECTRA, we also apply the encoder part in T5 and BART to be E-Extractive reader. The other one is using the encoder-decoder models where the decoder is to obtained the contextual representation of each token in the context in an autoregressive way (see §7.2). We use both BART and T5 PrLMs and term this kind of reader as **ED-Extractive reader**.

7.2 Generative Reader

We consider a generative reader consisting of an encoder and a decoder where the decoder is used to generate answers in an autoregressive way. Specially, the encoder takes a question \mathbf{q} and a context \mathbf{c} as input and outputs contextual representation \mathbf{h} . Then, the decoder takes the previously generated answer tokens as input and performs attention over \mathbf{h} and then generates the next token. Formally, given a tuple $(\mathbf{q}, \mathbf{c}, \mathbf{a})$, the training objective is to minimize the following loss function

$$\mathcal{L}_{\text{Gen}} = \sum_{i=1}^K \log \mathbf{P}(a_i \mid \mathbf{h}, a_{::i}) \quad (7.4)$$

where K is the number of tokens in answer \mathbf{a} , a_i is the i^{th} token in \mathbf{a} , and a_0 corresponds to a special beginning of sequence (`BOS`) token. In the inference time, we use the greedy search method to autoregressively generate the answer.

7.3 Compare Extractive and Generative Reader

7.3.1 Motivation

While both extractive and generative readers have been successfully applied to the Question Answering (QA) task, little attention has been paid toward the systematic comparison of them. Characterizing the strengths and weaknesses of the two readers is crucial not only for making a more informed reader selection in practice but also for developing a deeper understanding to foster further research on improving readers in a principled manner. Motivated by this goal, we make the first attempt to systematically study the comparison of extractive and generative readers for question answering. To be aligned with the state-of-the-art, we explore nine transformer-based large pre-trained language models (PrLMs) as backbone architectures. Furthermore, we organize our findings under two main categories: (1) keeping the architecture invariant, and (2) varying the underlying PrLMs. Among several interesting findings, it is important to highlight that (1) the generative readers perform better in long context QA, (2) the extractive readers perform better in short context while also showing better out-of-domain generalization, and (3) the encoder of encoder-decoder PrLMs (e.g., T5) turns out to be a strong extractive reader and outperforms the standard choice of encoder-only PrLMs (e.g., RoBERTa). We also study the effect of multi-task learning on the two types of readers varying the underlying PrLMs and perform qualitative and quantitative diagnosis to provide further insights into future directions in modeling better readers.

Model	In-domain Datasets							Out-of-domain Datasets						
	SQuAD	NewsQA	TQA	SQA	HQA	NQ	Avg.	DROP	RACE	BioASQ	TbQA	RE	DuoRC	Avg.
Single Task Learning														
T5 ED-Gen	90.75	71.65	79.61	86.21	79.89	78.04	81.02	48.08	48.89	67.36	60.30	84.94	61.35	61.82
BART ED-Gen	78.75	66.20	67.81	78.89	73.22	56.58	70.24	44.22	43.70	55.59	45.11	76.83	55.63	53.51
T5 E-Ext	92.47	72.63	76.09	<u>83.24</u>	80.67	80.00	<u>80.85</u>	53.14	52.06	71.26	61.92	85.78	62.80	64.49
BART E-Ext	92.19	<u>72.20</u>	73.12	77.19	80.61	<u>79.29</u>	79.10	51.57	48.82	<u>68.83</u>	51.29	86.04	<u>61.35</u>	61.32
ELECTRA	93.39	60.23	<u>76.31</u>	82.54	<u>80.99</u>	78.78	78.71	<u>55.43</u>	<u>49.80</u>	66.96	47.80	<u>86.23</u>	54.90	60.19
RoBERTa	<u>92.64</u>	59.95	72.97	81.62	81.21	78.95	77.89	55.88	47.72	64.47	52.31	86.69	55.75	60.47
Multi-Task Learning														
T5 ED-Gen	91.41 _{+0.66}	71.29 _{-0.36}	80.01 _{+0.40}	86.46 _{+0.25}	79.70 _{-0.19}	78.09 _{+0.05}	<u>81.16</u> _{+0.14}	51.20 _{+3.12}	49.66 _{+0.77}	68.72 _{+1.36}	62.90 _{+2.60}	85.84 _{+0.90}	63.76 _{+2.41}	63.68 _{+1.86}
BART ED-Gen	88.63 _{+0.88}	68.91 _{+2.71}	74.91 _{+7.10}	82.52 _{+3.63}	80.53 _{+7.31}	75.78 _{+19.20}	78.55 _{+8.31}	55.20 _{+10.98}	50.04 _{+6.34}	63.78 _{+8.19}	54.81 _{+9.70}	80.94 _{+4.11}	58.47 _{+2.84}	60.54 _{+7.63}
T5 E-Ext	92.84 _{+0.37}	73.51 _{+0.88}	<u>77.37</u> _{+1.28}	82.89 _{-0.35}	81.92 _{+1.25}	80.74 _{+0.74}	81.55 _{+0.70}	59.10 _{+5.96}	54.01 _{+1.95}	<u>71.13</u> _{-0.13}	64.90 _{+2.98}	86.53 _{+0.75}	65.01 _{+2.21}	66.78 _{+2.29}
BART E-Ext	92.46 _{+0.27}	<u>72.11</u> _{-0.09}	72.24 _{-0.88}	76.53 _{-0.66}	82.04 _{+1.43}	79.40 _{+0.11}	79.13 _{+0.03}	58.22 _{+6.65}	50.40 _{+1.58}	70.72 _{+1.89}	56.29 _{+5.00}	<u>86.79</u> _{+0.75}	61.95 _{+0.60}	64.06 _{+2.74}
ELECTRA	<u>93.27</u> _{-0.12}	60.59 _{+0.36}	72.96 _{-3.35}	82.03 _{-0.51}	83.10 _{+2.11}	79.16 _{+0.38}	78.52 _{-0.19}	<u>62.56</u> _{+7.13}	50.29 _{+0.49}	71.50 _{+4.54}	54.60 _{+6.80}	87.14 _{+0.91}	56.88 _{+1.98}	63.83 _{+3.64}
RoBERTa	<u>93.41</u> _{+0.77}	59.56 _{-0.39}	72.23 _{-0.74}	80.98 _{-0.64}	82.37 _{+1.16}	<u>79.55</u> _{+0.60}	78.02 _{+0.13}	64.47 _{+8.59}	<u>51.81</u> _{+4.09}	69.15 _{+4.68}	53.63 _{+1.37}	86.31 _{-0.38}	56.06 _{+4.31}	63.55 _{+3.11}

Table 7.1: Comparison of readers based on the different PrLMs by F1 Score. Inference length of T5 is full length of context, 512 for ELECTRA, and 1024 for BART and RoBERTa. TQA: TriviaQA; SQA: SearchQA; HQA: HotpotQA; NQ: NaturalQuestions; TbQA: TextbookQA; RE: RelationExtraction. Bold numbers denote for the best result and underline numbers for the second best.

7.3.2 Experiments and Results

Here, we present the comparison cross different PrLMs including standard encoder-only models for extractive readers.

The Selection of Each Model’s size We use the encoder in T5 large model for the T5 E-Extractive reader so that it is of similar size as RoBERTa and ELECTRA extractive readers ($\sim 330M$)¹. When using BART PrLMs for extractive reader, we only use BART E-Extractive reader but not ED-Extractive reader because the former performs better even though it has less parameters (204M) than the later one has larger size. T5 generative reader is also smaller (223M), but this is better than using T5 large generative reader to compare with others, which is way too larger than other readers (737M). For BART generative reader, it is larger than other readers (406M). One potential issue for the abovementioned setting is that even though we choose

¹Note that the T5 PrLM is already trained on SQuAD, while others do not. However, based on the results on SQuAD, T5 does not have advantage over other models on this dataset.

the best comparison setting, still each model size are different, and thus if a model perform inferior than others, it might due to the smaller model size. However, the following conclusion we draw does not effect by this issue.

Are Encoder-decoder PrLMs Good for Extractive Readers? Based on Table 7.1, we find that encoder-decoder PrLMs outperform encoder-only PrLMs as extractive readers on average. Both T5 and BART E-Extractive readers perform better than RoBERTa and ELECTRA on IID and OOD datasets under single- as well as multi-task learning regardless of less parameters of T5 and BART. This observation is exciting since instead of using standard encoder-only PrLMs for extractive reader, encoder-decoder PrLMs are actually better choice.

Which reader generalize better on OOD? The extractive reader generalize better on OOD datasets. In both single- and multi-task learning, T5 E-Extractive reader shows the best performance, especially beating the BART generative reader even though the latter one has more parameters. BART E-Extractive reader also generalize well on OOD, and it also beats the BART generative reader even though the former has less parameters than the later.

Which PrLM is the best? Based on Table 7.1, we see that T5 is the best among four PrLMs in both single- and multi-tasks learning scenario on IID as well as OOD datasets. We observe two advantages of T5 over other PrLMs. First, T5 is much better than ELECTRA and RoBERTa on NewsQA data. In both single- and multi-task learning, RoBERTa and ELECTRA achieve around 60% F1 score on NewsQA, while both T5 extractive and generative reader achieved higher than 70% F1 score, yielding more than 10% improvements. Second, T5 is better at long context dataset. In IID, TQA and SQA, T5 ED-Generative reader outperforms other readers at least

3.30% and 3.67% in single-task, 7.05% and 4.43% in multi-task learning. On OOD datasets, TbQA and DuoRC, T5 E-Extractive reader is better than others at least by 9.61% and 1.45% in single-task, 8.61% and 3.06% in multi-task. We would like to mention that this advantage of T5 is conditioned on using full inference length.

Which PrLM benefits more from Multi-task Learning? While multi-task learning is in general beneficial for all PrLMs, we find BART benefits the most from multi-task learning, especially for the generative reader. For example, on IID datasets. BART generative reader improves more than 8% on average while all other readers improves less than 1%. Similarly for OOD datasets, the improvement of multi-task learning on BART generative reader are more significant than other readers. To summarize,

1. Encoder-decoder PrLMs can be in fact used as extractive readers, they are even better than the conventional choice (encoder-only PrLMs) of extractive readers on average.
2. Extractive readers perform better than the generative readers on OOD datasets, especially for the ones based on the encoder-decoder PrLMs.
3. T5 is the best among four PrLMs since it performs better on the news domain and the long context. And the advantage of T5 is conditioned on using full inference length.
4. While in general multi-task learning turns out to be useful for all PrLMs, BART PrLM benefits the most.

7.3.3 Analysis

We investigate the behavior of extractive and generative models in long and short context and predicting answers which include rare characters.

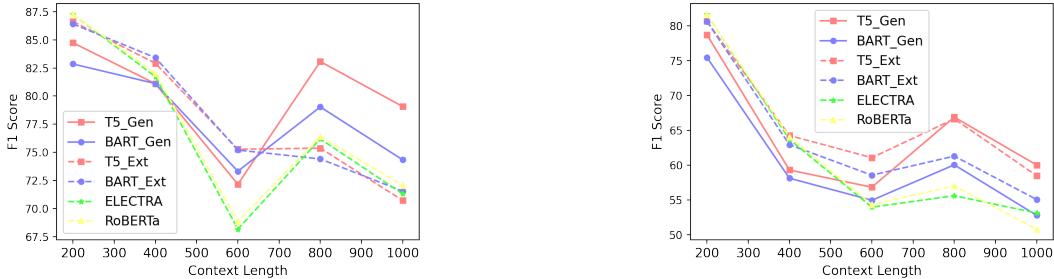


Figure 7.1: Comparison among generative and extractive readers on different length of the question and context. Left part for IID and right part for OOD datasets. Dash line for extractive and solid line for generative readers.

Long and Short Context As we discussed in previous section that generative readers have advantage over extractive counterparts. To further support this trend, we divide the testing sets into five subsets, where we count the total words in question and context, and choose five thresholds, 2/4/6/8/10 hundreds. It is worth to mention that since all extractive readers use the window-stride strategy (i.e. if the input length is longer than the maximum length, then the input is segmented into multiple inputs), so that the entire context is observable for extractive readers.

From Figure 7.1, we have two observations. First, on IID datasets, for questions and contexts with less than 600 words, the extractive ones always perform better than the generative ones (the dash lines are higher than the solid ones), but when the length are more than 600 words, the generative ones consistently outperform the extractive ones. This suggests that the extractive readers performs better in the short context while the generative readers perform better in long context. Second, on OOD datasets, T5 generative reader still presents advantage in the long context (more than 600 words), while BART generative reader performs worse than the extractive one in both short and long context. But the gap between the BART generative and extractive readers is less on the long context compared to the short context. It might suggest that the extractive reader has better generalization capacity than the generative one thus the advantage of generative reader in long context is weakened.

Rare Characters in Answer We find that some answers of testing sets include rare characters such as \acute{n} and \acute{t} (119 are found), thus we divide the testing sets into two subsets, one is the normal answer set where the answer does not have rare characters², the other one is with rare characters. The percentage of rare cases for IID and OOD datasets is 1.4% and 2%, respectively.

From Table 7.2, we have two observations. First, in normal case, the performance of extractive and generative readers are relatively comparable on both IID and OOD datasets, but in rare case, the extractive readers are better than the generative ones. This suggests that the extractive reader has better generalization than the generative ones. Second, we see that the rare tokens has worse impact on T5 than BART generative readers in both in- and out-of-domain datasets. Further investigation finds that 94 out of 119 rare characters can not be represented by T5 tokenizer (i.e. T5 tokenizer uses ‘junk $_c$ ’ special tokens to represent these characters), and tends to ignore these special characters in the generation time as the two examples shown in Table 7.3. Differently, BART tokenizer can represent all rare characters.

Improving generative readers performance in predicting rare answers is an important future work. To summarize,

1. Extractive readers performs better than the generative reader on short context, but generative one performs better on long context.
2. Generative readers performs worse in predicting answers with rare characters, and T5 performs worse than BART.

²Rare characters are any characters which are not belongs to the printable characters in the string library of Python. The printable characters include lower and upper case alphabets, digits, punctuation, and white-space.

Answer type	Domain	Gen		Ext			
		T5	BART	T5	BART	Ro	EL
Rare	IID	68.97	73.64	77.79	78.54	78.64	78.18
	OOD	59.25	79.84	85.22	84.95	80.73	86.94
Normal	IID	82.71	80.02	79.98	79.95	80.35	78.18
	OOD	68.28	64.19	69.9	66.91	67.75	68.12

Table 7.2: Compare extractive and generative readers in terms of rare and normal answers. Ro for RoBERTa and EL for ELECTRA.

Question	Answer	Prediction
Who was one of the most famous people born in Warsaw?	Maria Skłodowskacurie	Maria Skodowska-Curie
What museum preserves the memory of the crime?	Katyń Museum	Katy Museum

Table 7.3: Examples of questions with answers containing rare characters and the prediction of T5-Gen.

7.4 Discussion and Summary

We systematically compare the extractive and generative readers for QA tasks. Two sets of experiments are designed to control the effects of different PrLMs and the size of the models. By conducting experiments on 12 QA datasets, our findings provide guidelines on how to choose extractive or generative readers gave their strengths and weakness. While current work investigates the pros and cons of extractive and generative models systematically, there are some hyperparameters that might affect the model performance. For example, it is known that different prompts in the input affect generative model performance (Mishra *et al.*, 2021). Also, it is worth studying the OOD performance of models deeply. Gokhale *et al.* (2022) compares multiple

ways to improve the OOD performance of an extractive model on QA task, and how these methods affect generative models have not been well-studied yet. Meanwhile, most of the work including this work evaluates OOD performance by averaging the performance across multiple datasets, but as mentioned in (Mishra *et al.*, 2020), the evaluation should be more carefully designed. Also, Diagnosing the performance of each OOD dataset can provide more insights. For example, why models perform better on BioASQ dataset than most other datasets (see Table 7.1), while previous work has shown that it is hard to transfer general model to biomedical domain Luo *et al.* (2022c). Investigating the reason behind the observations and improving the generative and extractive models are interesting research questions for the future.

Chapter 8

GENERALIZATION AND ROBUSTNESS OF READER

Deep neural networks have emerged as a widely popular architectural choice for modeling tasks in multiple domains such as (but not limited to) computer vision Yuille and Liu (2021), natural language processing Hochreiter and Schmidhuber (1997); Vaswani *et al.* (2017b), and audio Hannun *et al.* (2014). While these models are highly capable of learning from training data, recent studies show that they are quite prone to failure on new test sets or under distribution shift Taori *et al.* (2020), natural corruptions Hendrycks and Dietterich (2019), adversarial attacks Goodfellow *et al.* (2015), spurious correlations Beery *et al.* (2018), and many other types of “unseen” changes that may be encountered after training. This shortcoming stems from the *i.i.d.* assumption in statistical machine learning which guarantees good performance only on test samples that are drawn from an underlying distribution that is identical to the training dataset. For instance, digit recognition models trained on the black-and-white MNIST training images are almost perfect ($> 99\%$ accuracy) on the corresponding test set, yet their performance on colored digits and real-world digits from street number plates is less than 75%. Similarly, state-of-the-art NLP models have been shown to fail when negation is introduced in the input Kassner and Schütze (2020). These findings pose a significant challenge to the practical adoption of these models and their reliability in the real-world.

To test model performance beyond the traditional notion of in-domain (ID) generalization, two prominent ideas have emerged: out-of-domain (OOD generalization) *a.k.a.* domain generalization¹, and adversarial robustness. The OOD generalization

¹In this paper we use these two terms interchangeably.

objective expects a model which is trained on distribution \mathcal{D} to perform reliably on unseen distributions $\mathcal{D}_e, e \in \{1, \dots, n\}$, that differ from \mathcal{D} . For a trained classifier f^* , OOD accuracy on previously unseen distribution \mathcal{D}_e is defined as:

$$\text{acc}_{\text{OOD}}^e = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}_e} [\mathbb{I}(f^*(\mathbf{x}) = \mathbf{y})] \quad (8.1)$$

To define adversarial robustness, consider an input \mathbf{x} and a true label \mathbf{y} . For a classifier loss function ℓ , a loss-maximizing perturbation δ^* within Δ_ϵ (an ϵ -bounded neighborhood of \mathbf{x}) is defined as:

$$\delta_{\mathbf{x}}^* = \max_{\delta \in \Delta_\epsilon} \ell(f^*(\mathbf{x} + \delta), \mathbf{y}). \quad (8.2)$$

The second idea is that of adversarial robustness. Recent work on adversarial examples has revealed the vulnerability of deep neural networks against small perturbations of the original data. Adversarial robustness in such under this setting is defined as the accuracy of the classifier on adversarial samples $\mathbf{x} + \delta_{\mathbf{x}}$, where the perturbation lies within an ℓ_p norm bound: $\|\delta_{\mathbf{x}}\|_p < \epsilon$.

$$\text{acc}_{\text{rob}} = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} \mathbb{I}(f^*(\mathbf{x} + \delta_{\mathbf{x}}) = \mathbf{y}). \quad (8.3)$$

In the context of text classification, the norm-bound can also be in the form of small character-level or word-level perturbations such as swapping, inserting, or deleting characters or words. In essence, adversarial robustness measures the invariance of the classifier to small perturbations of the input.

Various methods have been developed that either improve OOD generalization or improve adversarial robustness. Notable among these are techniques that modify the distribution of the training dataset. In this paper, we focus on three major data modification techniques – the use of additional datasets (also known as multi-source training), data augmentation, and data filtering; in addition we also consider

model-based debiasing techniques which do not alter the data distribution explicitly. In Gokhale *et al.* (2022), we study the performance of these methods on extractive question answering (QA).

8.1 Categorization of Domain Generalization Methods

In this section, we provide a categorization of methods that are typically used as baselines for domain generalization. We briefly explain the method and provide relevant related work in which these ideas are used as methods for domain generalization. Throughout this paper, we will refer to the original training distribution as the “*source*” and the out-of-distribution datasets as the “*targets*”.

Single-Source Training (**SS**) refers to the “vanilla” baseline which is trained only on the source dataset, without any dataset modification. **SS** utilizes no other information apart from the single source dataset \mathcal{D} and updates parameters θ of classifier f to minimize the risk on the source using approaches such as ERM (Vapnik and Chervonenkis, 1991).

$$\underset{\theta}{\text{minimize}} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} \ell(f(\mathbf{x}; \theta), \mathbf{y}). \quad (8.4)$$

Multi-Source Training (**MS**). This method is identical to **SS** except that additional training datasets \mathcal{D}' are used for risk minimization.

$$\underset{\theta}{\text{minimize}} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D} \cup \mathcal{D}'} \ell(f(\mathbf{x}; \theta), \mathbf{y}). \quad (8.5)$$

Usually \mathcal{D}' are designed for the same task as \mathcal{D} but may have different styles, characteristics, or sources of collection. For instance, while both SNLI (Bowman *et al.*, 2015a) and MNLI Williams *et al.* (2018a) are datasets for natural language inference with identical class labels, SNLI was collected from image captions, while MNLI was

collected from Open American National Corpus².

Gulrajani and Lopez-Paz (2020) provide an extensive comparative study of models trained for multi-source domain generalization for image classification and surprisingly find that if multiple source domains are available, ERM is empirically the best approach as compared to specially designed DG methods such as meta-learning Li *et al.* (2018a), learning domain-invariant features Ganin *et al.* (2016), invariant risk minimization Arjovsky *et al.* (2019), etc. These findings have also been observed on text classification experiments in Koh *et al.* (2021). Hendrycks *et al.* (2020a) show that pre-training transformer architectures on diverse data leads to higher OOD accuracies on multiple tasks such as semantic textual similarity, sentiment classification, reading comprehension and natural language inference.

Data Augmentation (DA). When additional training distributions are not directly available, transformations of samples in \mathcal{D} using pre-defined augmentation functions can be used to create \mathcal{D}' and train the model. Such data augmentation functions are typically derived from existing knowledge about the invariance of the task w.r.t. certain transformations. For instance, for image classification, addition of small noise, small translations, scaling, etc. are common data augmentation functions, since they do not change the true label for the image. Similarly, for text inputs, synonyms of words are commonly used since they do not change the semantics of the sentence. NLP data augmentation techniques include UDA Xie *et al.* (2020), EDA Wei and Zou (2019), and back-translation for question answering (Longpre *et al.*, 2019).

Data Filtering (DF). Dataset filtering has been previously explored for quality control, such as, removing noise and artifacts to curate and improve publicly sourced datasets. However, there has been recent interest in considering DF as a method for

²<https://www.anc.org/>

bias reduction and generalization. This idea can be traced back to work by Zellers *et al.* (2018, 2019), that proposed DF as an algorithmic method to avoid annotation artifacts and spurious correlations during dataset construction. AFLite Bras *et al.* (2020) extended this idea to a generic filtering methodology that can work without any pre-defined rules or strategies. Instead, AFLite operates by utilizing several weak learners (such as support-vector machines) trained over small subsets to identify samples that are easy to classify. It is argued that such samples are more likely to carry biases, and as such, could be removed. AFLite suggests that reduction of a dataset to even 10% of the original size can boost OOD accuracy on NLI. In the vision domain, similar ideas have been proposed concurrently, including REPAIR Li and Vasconcelos (2019) and RESOUND Li *et al.* (2018c), in which instead of completely removing samples, biased samples are assigned smaller weights. However these methods require a prior knowledge of the bias variable. Liu *et al.* (2021) have recently proposed a simple approach which upweights samples which have higher loss – this is shown to improve worst-group accuracy without having access to the bias variable.

Model De-biasing (DB). Methods under this category do not directly alter the training dataset, but instead resort to changes in the modeling technique – these changes can be in terms of the optimization function, regularization, additional auxiliary costs, etc. The main idea in DB is to utilize known biases (or identify unknown biases) in the data distribution, model these biases in the training pipeline, and use this knowledge to train robust classifiers Clark *et al.* (2019); Wu *et al.* (2020); Bhargava *et al.* (2021). In the image classification literature, there is growing consensus on enforcing a consistency on different views (or augmentations) of an image in order to achieve debiasing Hendrycks *et al.* (2020b); Xu *et al.* (2020); Chai *et al.* (2021); Nam *et al.* (2021). Unlike DF, model de-biasing does not directly alter the training

Method	In-Domain EM. (%)	OOD EM. (%)						
		DROP	RACE	BioASQ	TBQA	R.E.	DuoRC	Avg
SS	63.76	20.09	19.29	33.91	28.61	62.82	32.71	32.91
MS	65.07	26.88	27.45	45.01	40.52	72.86	43.44	42.69
DA	63.84	19.23	19.73	32.31	28.54	61.97	32.31	32.35
DB	64.58	20.83	19.73	34.64	31.20	63.64	35.98	34.34
DF	49.56	9.25	11.72	20.94	19.63	45.28	21.45	21.38

Table 8.1: QA Result: Source (IID) accuracy and domain generalization (OOD) on the Question Answering benchmark with NaturalQuestions as source dataset. EM: Exact-Match.

Method	Model Based #Queries	Model Free EM. (%)						
		CharSwap	EasyData	Embedding	WordNet	CheckList	CLARE	Avg
SS	19.55	60.29	52.17	61.21	58.41	63.22	61.92	59.54
MS	21.97	62.22	52.65	63.22	59.84	64.42	63.55	60.98
DA	21.91	60.88	54.52	62.02	59.82	63.42	62.36	60.5
DB	20.40	61.62	53.16	62.35	59.32	64.03	63.01	60.58
DF	19.19	47.97	42.48	48.55	47.19	49.34	48.72	47.38

Table 8.2: QA Result: Comparison of robustness in terms of model-based evaluation (number of queries needed to fool the model) and model-free (accuracy on adversarial transformations).

distribution, but instead allows the model to learn which biases to ignore.

8.2 Experiments and Results

We focus on extractive QA. Given a passage (or “context”) and a question, the task is to extract the answer span from the passage.

Methods. We use BERT Devlin *et al.* (2018) as the backbone model for each method. We use MRQA (Fisch *et al.*, 2019) which is a collection of 12 publicly

available multi-domain QA datasets – with Natural Questions (NQ) (Kwiatkowski *et al.*, 2019c) as the source dataset. SQuAD, NewsQA, HotpotQA, SearchQA, and TriviaQA are used as additional datasets for multi-source training. Similar to NLI, we use EDA for DA by applying EDA on the question. We apply the augmentation to all samples in the training set and combine them with the original set to train a DA model. For model de-biasing (DB), we use Mb-CR approach (Wu *et al.*, 2020), where a teacher and bias models are trained *a priori*, and are used for debiasing.

We modify AFLite for our QA task of span prediction, since AFLite was originally designed for classification tasks. To do so, we first randomly divide the training set into 10 subsets (or folds) $S_{1:10}$. For $k \in \{1, \dots, 10\}$, we pick S_k as the held-out test set, and train models on the rest, and obtain 10 such models. At test time, models are used for predicting an answer by only looking at the context (without access to the question) – this allows us to identify strong spurious correlations in the dataset. Based on the predictions, samples are sorted on the basis of their F1 score. A higher F1 score implies that the model is more likely to answer the question without even knowing the question. We retain 10% samples with the lowest F1 scores – these represent the task since the model is not likely to predict the correct answer without knowing the question.

Evaluation Protocol. We report exact-match (EM) accuracy for MRQA. To evaluate the generalization performance, we use six OOD development sets from MRQA: DROP, RACE, BioASQ, TextbookQA, RelationExtraction, and DuoRC. For robustness, we use the “Morphues” attack (Tan *et al.*, 2020) on the question as the model-based evaluation. Model-free methods are six pre-defined operations to transform question in the test inputs into adversarial examples. These six methods are: CLARE Li *et al.* (2021), character-swap Pruthi *et al.* (2019), Checklist Ribeiro *et al.*

(2020), EDA Wei and Zou (2019), counter-fitted embeddings (Emb) Alzantot *et al.* (2018).

Results. Table 8.1 shows the performance of each method in terms of in-domain and out-of-domain accuracy. We observe that two methods, **MS** and **DB**, improve the generalization performance on each out-of-domain dataset and also improve the in-domain performance. The improvement of **MS** is larger than **DB**. **DA** improves on some out-of-domain datasets but not all, and it also improves the in-domain performance. **DF** dramatically reduces both out-of-domain and in-domain datasets.

Table 8.2 shows that except for **DF**, all methods improve over **SS** for both model-based and model-free robustness evaluation. **MS**, **DA**, and **DB** improve the robustness in all transformations of model-free evaluation as well as the model-based evaluation, where **MS** achieves the best performance in model-based and model-free evaluation. **DF** significantly hampers the model-free robustness with drop in all transformations, meanwhile, the model-based robustness also drops.

8.3 Discussion and Summary

Recently, Miller *et al.* (2021) have empirically shown linear trends between in-distribution and out-of-distribution performance on multiple image classification tasks, across various model architectures, hyper-parameters, training set size, and duration of training. They also show that there are certain settings of domain shift under which the linear trend does not hold. Our work empirically shows that while data filtering may benefit OOD generalization on the NLI benchmark, this does not hold for other tasks such as image classification and question answering. This suggests that data filtering may benefit generalization in certain types of domain shift, but not on others. Concurrently, Yi *et al.* (2021) have theoretically shown that models robust to input

perturbations generalize well on OOD distribution within a Wasserstein radius around the training distribution. Our empirical observations agree with the theory of Yi *et al.* (2021).

To summary, we conduct a comprehensive study of methods that are designed for OOD generalization on extractive QA task. We evaluate each method on in-domain, OOD, and adversarial robustness. Our findings suggest that more data typically benefits both OOD and robustness. Data filtering hurts OOD accuracy and also hurts robustness. In the context of our findings and work by Miller *et al.* (2021); Yi *et al.* (2021), we recommend that methods designed either for robustness or generalization should be evaluated on multiple aspects and not on the single metric that they are optimized for.

Chapter 9

TEXT READER: SELECT BEFORE YOU ANSWER

Open book question answering (OBQA) requires a system to find the relevant documents to reason the answer to a question. It has wide and practical Natural Language Processing (NLP) applications such as search engines (Kwiatkowski *et al.*, 2019a) and dialogue systems (Reddy *et al.*, 2019; Choi *et al.*, 2018). Among several OBQA datasets (Dhingra *et al.*, 2017; Mihaylov *et al.*, 2018; Khot *et al.*, 2020), HotpotQA Yang *et al.* (2018a) is more challenging because it requires a system not only to find the relevant passages from large corpus but also find the relevant sentences in the passage which eventually reach to the answer. Such a task also increases the interpretability of the systems.

To address this challenge, most of the previous work (Nie *et al.*, 2019; Fang *et al.*, 2020; Tu *et al.*, 2019; Groeneveld *et al.*, 2020) use two-step pipeline: identify the most relevant passage by one model and then match each question with a single sentence in the corresponding passage by another model. Such systems are heavy in terms of the size of the models which requires long training and inference time. Green AI has recently been advocated to against the trend of building large models which are both environmentally unfriendly and expensive, raising barriers to participation in NLP research Schwartz *et al.* (2020). Apparently, systems using multiple models to solve HotpotQA task do not belong to the family of Green AI. Furthermore, the benefits of learning from passage ranking and selecting relevant sentences are not well utilized by these systems. Intuitively, if a passage is ranked high, then some sentences in the passage should be selected as relevant. On the other hand, if a passage is ranked low, then all sentences in the passage should be classified as irrelevant.

Question: The football manager who recruited David Beckham managed Manchester United during what timeframe?

Passage1, 1995–96 Manchester United F.C. season:

The 1995-96 season was Manchester United's fourth season in the Premier League, and their 21st consecutive season in the top division of English football. United finished the season by becoming the first English team to win the Double (league title and FA Cup) twice. *Their triumph was made all the more remarkable by the fact that Alex Ferguson had sold experienced players Paul Ince, Mark Hughes and Andrei Kanchelskis before the start of the season, and not made any major signings. Instead, he had drafted in young players like Nicky Butt, David Beckham, Paul Scholes and the Neville brothers, Gary and Phil.*

passage2, Alex Ferguson: *Sir Alexander Chapman Ferguson, CBE (born 31 December 1941) is a Scottish former football manager and player who managed Manchester United from 1986 to 2013.* He is regarded by many players, managers and analysts to be one of the greatest and most successful managers of all time.

Answer: from 1986 to 2013

Supporting facts: [[{"1995-96 Manchester United F.C. season": 2}, {"1995-96 Manchester United F.C. season": 3}, {"AlexFerguson": 0}]]

Figure 9.1: An example from the HotpotQA dataset, where the question should be answered by combining supporting facts(SP) from two passages. In the SP, the first string refers to the title of passage, and the second integer means the index of the sentence.

To build a Green AI system and take advantage of multi-task learning, we introduce a Two-in-One model in (Luo *et al.*, 2022a), a simple model trained on passage ranking and sentence selection jointly. More specifically, our model generates passage representations and sentence representations simultaneously, which are then fed to a passage ranker and sentence classifier respectively. Then we promote the interaction between passage ranking and sentence classification using consistency and similarity constraints. The consistency constraint is to enforce that the relevant passage includes

relevant sentences, while the similarity constraint ensures the model to generate the representation of relevant passages more closer to the representations for relevant sentences than irrelevant ones. The experiments conducted on the HotpotQA datasets demonstrate that our simple model achieves competitive results with previous systems and outperforms the baselines by 28%.

9.1 Method

We aim to jointly conduct two tasks, passage ranking and supporting facts selection for HotpotQA. Given a question Q , the goal is to simultaneously rank the set of candidates $A = \{a_1, \dots, a_i\}$ and identify the supporting facts for the TopK¹ passages.

9.1.1 Model: Two-in-One Framework

We introduce the proposed joint model for passage ranking and support fact selection, Two-in-One, which uses state-of-the-art transformer-based model (Vaswani *et al.*, 2017c) to encode questions and contexts. In this work, we use RoBERTa (Liu *et al.*, 2019b), however, any other variants like ELECTRA (Clark *et al.*, 2020a) can be applied in this framework. The model architecture is given in Figure 9.2. On top of the encoder, there are two MLP layers to score passages and sentences respectively. In details, given a question and a passage, we firstly create an input to feed through RoBERTa Liu *et al.* (2019b) by concatenating the question and the passage as follows, $\langle s \rangle Q \langle /s \rangle S_1 \langle /s \rangle S_2 \dots \langle /s \rangle S_k \langle /s \rangle$ where $\langle s \rangle$ and $\langle /s \rangle$ are special tokens in RoBERTa, S_i is the i^{th} sentence from a passage. We take $\langle s \rangle$ as the contextual representation for passage ranking and the $\langle /s \rangle$ in front of each sentence for sentence selection. The passage ranker and the sentence classifier have identical structure (two-layer Multiple-Layer Perceptron(MLP)) but different weights.

¹The value of K depends on the task, and for HotpotQA, K is 2.

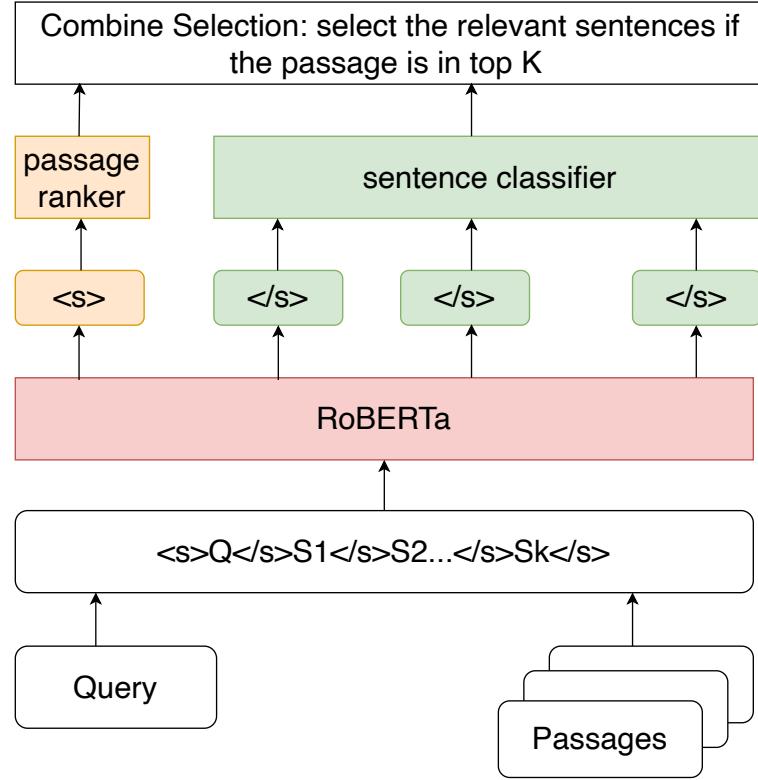


Figure 9.2: The architecture of Two-in-One model for passage ranking and relevant sentence selection. For HotpotQA dataset, K is two.

The model is jointly trained by passage loss and sentence loss. In detail, during the training time, we assign the relevant passages and sentences with ground truth score 1 while irrelevant passages and sentences with ground truth score -1. Then, Mean Square Error(MSE) loss is applied to calculate the passage and sentence loss as follows,

$$\begin{aligned}\mathcal{L}^{pass} &= (\hat{y} - y)^2, \\ \mathcal{L}^{sent} &= \sum_{i=1}^K (\hat{x}_i - x_i)^2, \\ \mathcal{L}^{joint} &= \mathcal{L}^{pass} + \mathcal{L}^{sent},\end{aligned}\tag{9.1}$$

where \hat{y} is the predicted passage score, y is the ground truth score of the passage, \hat{x}_i and x_i are the predicted sentence score and ground truth score of S_i , respectively, and

K is the total number of sentences in the passage. We simply sum up the passage loss and sentence loss to jointly update model parameters.

During the inference time, passages are ranked based on the logits given by the passage ranker. For the sentence classification, we take 0^2 as the threshold to classify the relevance of each sentence: if the score given by the sentence classifier is larger than 0, then it is relevant; otherwise, irrelevant.

Next, we introduce two constraints to facilitate the interaction between these two tasks.

9.1.2 Consistency Constraint

Intuitively, if a passage is relevant to the question, then there are some sentences from the passages that are relevant; on the other hand, if a passage is not relevant to the answer, then there should not be relevant sentences inside the passage. Thus, we propose a consistency constraint over the passage ranker and sentence classifier to minimize the gap between the passage score and the maximum sentence score. The loss function is as follows:

$$\mathcal{L}^{con} = (\hat{y} - \max(\mathbf{x}))^2, \quad (9.2)$$

where $\mathbf{x} = [\hat{x}_1 \dots \hat{x}_n]$ denotes a stack of predicted sentence scores.

9.1.3 Similarity Constraint

As we have shown at the beginning of this section, token $\langle s \rangle$ is used to get the passage score, and each token $\langle /s \rangle$ is used to get the sentence score. Intuitively, the similarity between token $\langle s \rangle$ of a relevant passage is more close to token $\langle /s \rangle$ of a relevant sentence than to $\langle /s \rangle$ of any irrelevant sentence. To enforce this constraint,

²The reason for threshold “0” is that it is the middle value of 1 and -1, which are labels for relevant and irrelevant sentences in the training time.

Model	# Parameters	SP Precision	SP Recall	SP F1	SP EM	Passage EM
Sentence Selection Baseline	~330M	67.96	81.05	72.02	28.12	69.70
Passage Selection Baseline	~330M	66.43	56.55	60.20	27.30	90.44
Two-in-One + sim (Ours)	~330M	88.06	85.68	85.82	59.17	91.11
QUARK	~1020M*	N/A	N/A	86.97	60.72	N/A
SAE(RoBERTa)	~660M+*	N/A	N/A	87.38	63.30	N/A
HGN(RoBERTa)	~330M+*	N/A	N/A	87.93	N/A	N/A

Table 9.1: The Results for two baselines and Two-in-One model with similarity constraint on dev set of HotpotQA distracting dataset. SP stands for supporting facts and EM for Exact Match. * refers to estimation. The bottom systems have much larger model size than our method, where QUARK, is the result of a framework with 3 BERT models, SAE uses two large language models and an GNN model, and HGN uses a large language model, a GNN model and other reasoning layers.

we use triplet as follows:

$$\mathcal{L}^{sim} = \frac{1}{N \cdot M} \sum_{i=1}^N \sum_{j=1}^M (\max\{d(v^p, v_i^r) - d(v^p, v_j^n) + m, 0\}), \quad (9.3)$$

where $d(\cdot, \cdot)$ is the Euclidian similarity, N is the number of relevant sentences, M is the number of irrelevant sentences, v^p, v^r, v^n is the vector representation of the relevant passage, relevant sentence, and irrelevant sentence respectively. Equation 9.3 enforces that all the relevant sentences should have higher similarity with the passage than all the irrelevant sentences by a margin m ; otherwise, the model would be penalized. In practice, we set the margin m at 1 and find optimum results. We train our model in an end-to-end fashion by combining \mathcal{L}^{joint} , \mathcal{L}^{con} and \mathcal{L}^{dis} .

9.2 Experiment and Results

In this section, we first describe the training setup, and then introduce two baselines. We evaluate the two baselines and our proposed joint model on the HotpotQA dataset. Yang *et al.* (2018a) provides two metrics for supporting facts evaluation, exact matching (EM) and F1 score. We also present the precision and recall of SP, and the exact

matching of passages for detailed comparison. We mainly compare our model with the QUARK system Groeneveld *et al.* (2020) since both QUARK and our method simply use language models without involving complicated reasoning models. For reference, we also present other state-of-the-art models in Table 9.1. Lastly, we conduct an ablation study to show the effectiveness of the proposed similarity loss and consistent loss.

Experiment Setup We use Huggingface and Pytorch (Paszke *et al.*, 2019) libraries to implement each model. We use 4 TX1080 and V100 NVIDIA to train models in 5 epochs with a learning rate of 1e-5, batch size of 32. We set the maximum input length in training to be 512.

Baseline To have comparable size of the model, two baselines have similar structure as our Two-in-One model. Our model has two classification heads, whereas each of the baselines has one classification head. One baseline is to select relevant sentences, and the other one is to rank passages.

Sentence Selection Baseline The first baseline is to select relevant sentence, and particularly, we use a RoBERTa-large with an additional MLP trained on question and a single sentence: $\langle s \rangle Q \langle /s \rangle S \langle /s \rangle$, where Q is a question and S is a sentence. Although this model can not predict the relevant passage directly, based on the assumption that relevant passages include relevant sentences, we pick up two relevant passages based on the top2 sentence scores. When the top1 and the top2 sentences are from the same passage, we continue searching based on the ranking sentence scores until the second document comes up. Then the supporting facts are those sentences from the relevant documents with a score larger than 0.

Passage Selection Baseline In the second baseline, again, we use RoBERTa-large but with the goal of passage selection. The input to the model is a question and a passage: $\langle s \rangle Q \langle /s \rangle P \langle /s \rangle$. Since such a model can not predict sentence relevancy score, based on the statistic of HotpotQA that majority of training set has two supporting facts and the most of them are the first sentences in a paragraph, we select supporting facts by the first sentence of the top1 and top2 passages.

Result As we see from Table 9.1, Two-in-One framework outperforms two baselines with large-margin improvement in all metrics, especially we see a significant improvement on the EM of SP. Our framework outperforms the Sentence Selection Baseline by 20% and 4.5% improvement on the precision and recall of SP, respectively, which demonstrates that jointly learning is beneficial for sentence classification. Also, jointly learning benefits for the passage ranking by comparing Two-in-One with Passage Selection Baseline on the EM of passage. Besides, we also compare Two-in-One with QUARK Groeneveld *et al.* (2020), a framework involving three BERT models, (roughly three times larger than ours). Two-in-One achieves comparable results in terms of F1 and EM of SP regardless of much less parameters in our system. Notice that we do not have the other three values because they are not presented in their original paper.

Ablation To evaluate the impacts of the consistency constraint and the similarity constraint, we conduct experiments with and without constraints. From Table 9.2, we see that both consistency constraint and similarity constraint improve F1 and EM of SP and the similarity constraint also improves the EM of passages. We found that without any constraint, though the model can rank the passages well, it suffers from distinguishing between close sentences. The similarity constraint addresses this issue in some sense by maximizing the distance between relevant and irrelevant sentences.

Model	SP F1	SP EM	Passage EM
Two-in-One	85.52	58.67	90.93
Two-in-One + con	85.55	58.98	90.29
Two-in-One + sim	85.82	59.17	91.11
Two-in-One + con + sim	85.63	58.74	90.78

Table 9.2: The results for Two-in-One model with or without consistency and similarity constraints.

To better understand the impact of consistency constraint, we analyze the consistency between the passage score and the sentence score. The prediction of a model is consistent if the passage score agrees with the sentence scores and the agreement can be measured by the gap between the passage score and the maximum sentence score among all sentences in that passage. We observe that by adding the consistency constraint, the gap between the passage score and the sentence score is much smaller than without the consistency constraint, i.e. 0.03 v.s. 0.11. It demonstrates that the constraint is beneficial for consistent prediction.

9.3 Discussion and Summary

Model Architecture It is easy to extend the Two-in-One model to Three-in-One model such that besides the passage ranking and sentence selection modules, a third module can predict the answer span. Like the simple extractive QA model based on RoBERTa, where a linear layer or an MLP can predict the start and end position of the answer span. A restricted inference procedure can be enforced that the answer span should be predicted from the selected sentence given by the previous model. One benefit is to reduce the difficulty for the answer selection model since less sentences will be seen by the model and the second benefit is to increase the interpretability of the model. On the other hand, if the sentence selection model makes mistakes,

then such errors will carry to the answer span model which yields the wrong answer eventually.

Apply to Full Open Domain Setting We only study the distracting setting of HotpotQA in this work, where 10 passages are already given for each question. On the other hand, in the full open domain setting, the passages need to be chosen from a large corpus. A simple approach to adapt Two-in-One model to the later setting is to use a retriever Robertson *et al.* (2009); Karpukhin *et al.* (2020a); Luo *et al.* (2022c) to select the 10 passages and ask Two-in-one to choose the right passages and supporting facts.

Summary We present a simple model, Two-in-One, to rank passage and classify sentence together. By jointly training with passage ranking and sentence selection, the model is capable of capturing the correlation between passages and sentences. We show the effectiveness of our proposed framework by evaluating the model performance on the HotpotQA datasets, concluding that jointly modeling passage ranking and sentence selection is beneficial for the task of OBQA. Compared to the existing QA systems, our model, with fewer parameters and more green than previous models, can achieve competitive results. We also propose multiple future directions to improve our model such as exploring the relationship among passages, supporting sentences, and answers in modeling and generalizing our method on more datasets.

Chapter 10

LOGICAL REASONING OF READER

Logical Reasoning (LR) is one of the oldest topics and challenging task in AI community. LR plays fundamental roles in many domains including but not limited to math, science, laws, planning, and action reasoning (Banerjee *et al.*, 2020). While we have recently observed tremendous progress made in natural language processing (NLP), evident in large pretrained language models (PrLMs) have achieved superhuman performance on a number of NLU benchmarks, there is still much to explore regarding logical reasoning in natural language.

The earliest effort of logical reasoning mainly focus on designing formal logic language to represent rules and knowledge and develop automatic theorem prover to inference new facts. However such paradigm requires expert knowledge (about the formal logic syntax and semantics) and human effort to write the rules explicitly, and still suffers from scale-up issue in real applications. To ground the application of using logical reasoning, recent effort gradually shift to using neural networks to do logical reasoning, especially with PrLMs. Recent researchers create logical reasoning dataset by synthetically generating dataset based on first order logic rules and natural language templates, and such dataset maintain the structure of the logical reasoning but does not exhibit real world meaning (e.g. the rules are not true in real life). They show that PrLMs can be a “soft-reasoner”. On the other hand, some work also shows that such models have poor generalization capacity on data of unseen distribution.

To study the logical reasoning capacity of language models, we first introduce LogiGLUE benchmark which can serve as generalization test-bed in two aspects. First we collects 8 datasets spanning different tasks, multiple choice question an-

swering (MCQA), natural language inference (NLI), and fact verification (FV). With LogiGLUE, we study two research questions that are important, however, not well-studied. The first question is “*if there is a single model that generalizes well to different logical reasoning tasks?*”. To answer this question, we use the RoBERTa model (Liu *et al.*, 2019a) and encode the input text in three different formats to predict the answers. The second question is “*is there a correlation between commonsense reasoning and logical reasoning?*”. We study this question for two reasons: 1) both types of reasoning are considered as difficult, 2) these two tasks are sometime entangle with each others. To answer this question, we fine-tune the Unicorn model (Lourie *et al.*, 2021), a generative model trained on Rainbow (six commonsense reasoning datasets), on LogiGLUE to investigate if commonsense reasoning ability is beneficial for logical reasoning. We also train a sequence-to-sequence model on LogiGLUE and obtain a model called LogiT5, then further fine-tune LogiT5 on Rainbow datasets to investigate if logical reasoning is beneficial for commonsense reasoning.

Experimental results show that 1) while the language model is the same, task formalization plays a huge role in logical reasoning tasks; 2) model with commonsense reasoning skill outperforms model without commonsense knowledge on logical reasoning tasks; on the other hand, a model with logical reasoning skill only shows marginal benefits than models without logical reasoning skill on commonsense tasks. In the end, we also study the zero-shot and few-shot in-context learning performance of GPT-3 on LogiGLUE, and GPT-3 shows superior logical reasoning capacity given its larger model size.

10.1 Logical Reasoning Benchmark: LogiGLUE

We introduce LogiGLUE, a suite of natural language logical reasoning benchmarks with 8 datasets that cover different types of logical reasoning. In addition, LogiGLUE

Dataset	Train size	Test size	Synthetic	Task Type
ARCT	1,210	444	✗	MCQA
ReClor	4,638	500	✗	MCQA
LogiQA	7,376	651	✗	MCQA
TaxiNLI	10,032	7,727	✗	NLI
LogicNLI	16,000	2,000	✓	NLI
RuleTaker	69,762	20,192	✓	FV
Rulebert-Chain	56,000	9,334	✓	FV
Rulebert-Union	210,000	60,000	✓	FV

Table 10.1: Statistics of In-domain (IID) and out-of-domain (OOD) datasets of LogiGLUE benchmark.

includes three task formats, multiple choice question answer (MCQA), natural language inference (NLI), and fact verification (FV). Table 10.1 shows the statistics.

ARCT Habernal *et al.* (2018) is an argument reasoning comprehension task that given an argument with a claim and a premise, the goal is to choose the correct implicit warrant from two options.

ReClor Yu *et al.* (2019) is a multiple choice question answering task requiring logical reasoning extracted from standardized graduate admission examinations.

LogiQA Liu *et al.* (2020) is a multiple choice question answering dataset sourced from publicly available logical examination papers for reading comprehension that cover categorical reasoning, conditional reasoning, disjunctive reasoning, and conjunctive reasoning.

TaxiNLI Joshi *et al.* (2020) annotate partial MNLI dataset Williams *et al.* (2018b) with 10 logical reasoning skills.

RuleTaker Clark *et al.* (2020b) is a synthetic dataset generated from templates and simple first-order logic (FOL) rules with three connectives, implication, conjunction, and negation.

LogicNLI Tian *et al.* (2021) is a semi-synthetic NLI dataset that covers numerical reasoning, coreferential reasoning, abductive reasoning, and pragmatic reasoning. LogicNLI is generated by FOL rules and template language with human revision. Besides the convention of three labels in the NLI task, LogicNLI includes self-contradiction labels, meaning that both the hypothesis and the negation of the hypothesis follow the premise.

Rulebert Saeed *et al.* (2021) is a synthetic dataset generated using soft Horn rules mined from large RDF knowledge bases (Galárraga *et al.*, 2015). We leave out the probability and include two subsets of the dataset: **Chain-rule** which requires a sequence of reasoning steps; **Union-rules** which has five rules in the context.

Unique Format. We convert each dataset into a unique format such that each instance has a question, a context, and a set of answer choices. For MCQA tasks, the question, the context, and a set of answer choices are straightforward. For FV tasks, the statement is taken as the question, and two choices are given, true and false. NLI tasks are similar to FV tasks, but the answer choices are natural, contradictory, and entailment (self-contradiction is given to the LogiNLI dataset.)

10.2 Classification Models For Logical Reasoning Tasks

We use RoBERTa as the language encoder, and study three classification types of prediction. In general, all models have one encoder to produce the embeddings of the input context and a classification layer (multiple-layer perceptron) to produce the final answer. We testify three standard classification methods which are used to solve different NLU tasks in previous work.

Single Classification. This model takes a single string as input consisting of a context, a question, and all answer choices, and the classification layer takes the first

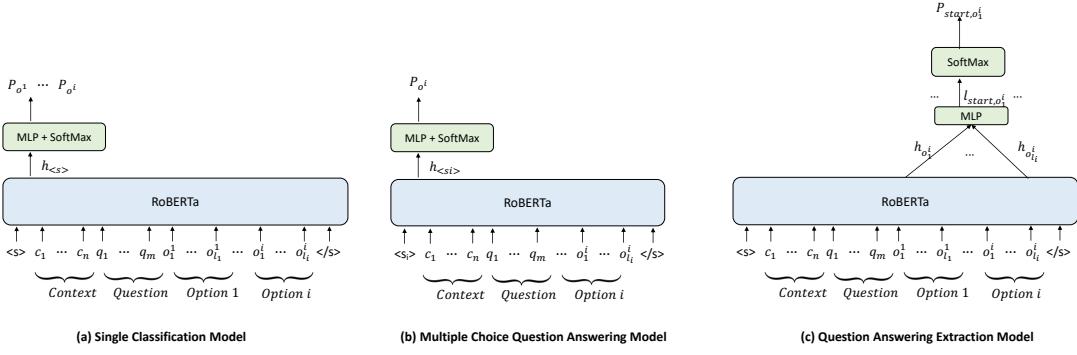


Figure 10.1: Three types of classification models: Single Classification: all answer choices are concatenated in one single input, and the model classifies one label from answer choices. Multiple Choice Question Answering: each answer is independently given to the model and the model predicts a score for each answer. For the answer extraction model, it classifies the start and end tokens from the answer choices (the classification layer of the end token is not shown in the figure).

Model	LogiGLUE								Avg.
	ARCT	ReClor	LogiQA	TaxiNLI	LogicNLI	RuleTaker	Rulebert-chain	Rulebert-union	
RoBERTa (SC)	<u>80.63</u>	27.00	27.50	92.14	85.30	99.50	99.59	99.88	76.44
RoBERTa(MCQA)	87.16	68.60	41.63	92.18	<u>83.25</u>	99.48	99.66	100.00	84.00
RoBERTa(EXT)	79.50	<u>59.60</u>	32.87	60.29	69.20	97.92	99.38	98.02	74.60
Previous SOTA	56.00	67.20	41.80	92.30	68.30	99.70	83.10*	100.00	—

Table 10.2: Results of single classification (SG), multiple-choice question answering (MCQA), and answer extraction (EXT) models on the LogiGLUE Benchmark. The best performance is highlighted in bold, and the second best is in underline. * denotes F1 score.

token of the input string ([CLS]) and produces a label from the number of given answer choices. The left model in Figure 10.1 shows the structure.

Multiple Choice Question Answering This model takes n inputs where n is the total number of the answer choice, and each input consists of a context, a question, and an answer choice, the classification layer again takes the first token of the input string to produce a score for each answer choice, and the final label is the answer choice which has the highest score. The middle model in Figure 10.1 shows the structure.

Question Answering Extraction has similar input as the single classification model, but the classification layer takes every token in the answer choice as input and produces a probability for it being the start token and a probability for being the end token. Eventually, the answer is extracted by the span between tokens with the highest start token probability and the highest end token probability. The right model in Figure 10.1 shows the structure.

10.3 LogiT5: A generative Model for Logical Reasoning Tasks

In the preceding section, we presented three classification models' performance. However, there has been a recent surge of interest in generative models due to their greater versatility compared to classification models. Generative models, for instance, can perform text generation tasks like summarization and storytelling, which are beyond the capabilities of classification models.

Therefore, we train a generative model, T5-large, on LogiGLUE, with the intuition that training the model on multi logical reasoning tasks will enhance the logical reasoning capacity of a model. We term such a model as LogiT5.

10.4 Experiments and Result

10.4.1 Performance of Classification Models

Table 10.2 shows the performance of three classification models based on RoBERTa-large. We find that while using the same language model and the training data, how to predict the answer causes a significant difference in performance. More precisely, MCQA models achieve the best performance on average, and significantly better than the other two models ($\sim 8\%$ better). On the three datasets, ARCT, ReClor, and LogiQA, MCQA models have obvious advantages over the other two types of models.

Model	LogiGLUE								Avg.
	ARCT	ReClor	LogiQA	TaxiNLI	LogicNLI	RuleTaker	Rulebert-chain	Rulebert-union	
T5	71.40	39.60	20.89	92.13	70.70	99.10	99.67	99.99	74.18
LogiT5	81.08	60.40	41.94	92.12	80.35	97.79	99.07	99.99	81.59

Table 10.3: LogiT5 achieves better performance than T5, demonstrating the benefits of training on a collection of logical reasoning tasks.

Also, MCQA models also achieve comparable performance on NLI and FV tasks. This suggests that MCQA is a unified model for addressing these three tasks. The MCQA models also achieve similar performance to SOTA models.

10.4.2 Performance of a Generative Model LogiT5

We devised two different settings: the first involved training a vanilla T5 model on each individual task in LogiGLUE, while the second setting training LogiT5 on each task in LogiGLUE. The results of these two models are compared in Table 10.3.

We observe that LogiT5 performs much better on average than the vanilla T5 model on the LogiGLUE benchmark, with a significant improvement in the low-resource tasks such as ARCT, ReClor, and LogiQA. However, the performance of both models on the remaining tasks is similar. These observations suggest that multi-task learning is beneficial for low-resource domains. Furthermore, we note that generative models are not as proficient as classification models in logical reasoning tasks. This indicates that treating logical reasoning as a classification task is relatively easier than approaching it as a text generation task.

10.4.3 Logical Reasoning and Commonsense Reasoning

To investigate whether commonsense knowledge is beneficial for logical reasoning, we fine-tune the Unicorn model Lourie *et al.* (2021) on each dataset in LogiGLUE.

Model	LogiGLUE								Avg.
	ARCT	ReClor	LogiQA	TaxiNLI	LogicNLI	RuleTaker	Rulebert-chain	Rulebert-union	
T5	71.40	39.60	20.89	92.13	70.70	99.10	99.67	99.99	74.18
Unicorn	83.56	50.60	38.86	89.16	74.25	95.79	98.93	99.99	78.89

Table 10.4: Results of transfer Unicorn model, a model with commonsense reasoning capacity, to LogiGLUE benchmark. The commonsense reasoning is beneficial for the logical reasoning.

Model	Rainbow						Avg.
	aNLI	Cosmos QA	HellaSWAG	Physical IQa	Social IQa	WinoGrande	
T5	77.42	80.13	81.57	79.87	72.83	74.74	77.76
LogiT5	76.63	79.83	81.43	81.18	73.08	78.30	78.41

Table 10.5: Results of transfer LogiT5 model, a model trained on LogiGLUE benchmark, to Rainbow benchmark. The logical reasoning shows little benefit for the commonsense reasoning.

Unicorn is a T5 model further fine-tuned on a collection of commonsense reasoning datasets and thus considered to contain rich commonsense reasoning knowledge. To compare, we fine-tune the vanilla T5 model on each dataset in LogiGLUE as baselines. As shown in Table 10.4, fine-tuned Unicorn model achieves better performance than the baseline, especially on ARCT, ReClor, and LogiQA, suggesting commonsense knowledge indeed helps logical reasoning significantly. We hypothesize the reason is that some commonsense reasoning questions involve multiple commonsense knowledge, and there are logical connections between them to reach the answer.

To investigate whether logical reasoning helps models to perform commonsense reasoning tasks, we fine-tune LogiT5 on each dataset in Rainbow. To compare, we fine-tune the vanilla T5 model on each dataset in Rainbow as baselines. As shown in Table 10.5, logical reasoning knowledge helps models perform commonsense reasoning tasks but only marginally.

Model	ARCT	ReColr	LogiQA	TaxiNLI	LogicNLI	Avg
Zero-shot	100.00	40.00	80.00	60.00	80.00	72.00
Def + pos(1)	60.00	20.00	60.00	60.00	40.00	48.00
Def + pos(k)	80.00	40.00	53.33	80.00	53.33	61.33

Table 10.6: Results for prompt learning on LogiGLUE using GPT-3.

10.4.4 GPT-3 Performance

To evaluate the performance of GPT-3, we randomly selected twenty samples for evaluation from each dataset such that selected samples represent diverse classes. We evaluate three prompting strategies: (1) **Def** where only task definition is provided, (2) **Def + pos(1)** where one positive example is provided along with task definition, and (3) **Def + pos(k)** where definition and k positive example are provided along with task definition to the model. Since all tasks in LogiGLUE are classification tasks, we have included examples corresponding to each class (i.e., k examples for **Def + pos(k)** prompting method). For statistical significance, we report the average score over 3 sets of randomly selected in-context examples for **Def + pos(k)**.

Table 10.6 shows the result in terms of accuracy. We find that GPT-3 achieves comparable and sometimes even better performance than fine-tuning results. In particular, GPT-3 achieves better performance on ARCT and LogiQA than the fine-tuning results. Surprisingly, we see that the performance of **Def** prompting method is better than other methods on 4 out of 5 datasets.

10.5 Discussion and Summary

Motivated by the importance of logical reasoning, we introduce the first logical reasoning multi-task benchmark. Our benchmark, called LogiGLUE, covers 8 datasets across three different tasks and diverse reasoning types. We then study how language

model performance on LogiGLUE, including three classification models, and find that multiple choice question answering model performs much better than other two types of models. We also investigate the relation of logical reasoning and commonsense reasoning and find the latter has a positive impact on the former. Lastly, we find that GPT-3 exhibits great logical reasoning and even performs better than full fine-tuning models. We hope that our benchmark can serve as the test-bed of logical reasoning of a system.

Chapter 11

MULTIMODAL READER: READING COMPREHENSION FROM TEXT AND IMAGE

Previous section focus on MRC from text, however, similar to information retrieval, multimodal MRC has increasing application in modern life and thus, more research attention has been drawn toward multimodal MRC. One of the important task is visual question answering. In the following, we mainly address two challenges in multimodal MRC, the evaluation and model.

11.1 Visual Question Answering

11.1.1 Existing Reader

Current state-of-the-art VQA systems are classification models (Tan and Bansal, 2019a; Li *et al.*, 2019; Gokhale *et al.*, 2020b,a; Banerjee *et al.*, 2021c), where a list of answer candidates are pre-defined (from the training set), i.e., a fixed answer vocabulary, then a model classifies one of the answers as the final prediction.

11.1.2 Proposed Reader

Classification Reader

We build a reader similar to existing reader but incorporate external knowledge. In particular, given a question, an image, and a piece of knowledge, we first concatenate the question with the knowledge and then apply a cross-modality model to encode the text with the image and generate a cross-modal representation. We feed this representation to a Multiple Layer Perceptron (MLP) which finally predicts one of

the pre-defined answers. We apply Cross-Entropy Loss to optimize the model. In this work, we use LXMERT (Tan and Bansal, 2019a), while any other cross-modality models like VisualBERT(Li *et al.*, 2019) can be adapted.

Proposed Extractive Reader

The classification model fails to generalize to out-of-domain answers, i.e., questions whose answers are not in the pre-defined answer vocabulary. To tackle this issue, we use an extraction model which is adapted from machine reading comprehension model (Chen *et al.*, 2017b; Karpukhin *et al.*, 2020a). The model extracts a span (i.e., a start token and an end token) from the knowledge to answer the question. The image caption is given to the model as well to incorporate the image information. We also inject a special word “unanswerable” before the caption so that the model can predict “unanswerable” if the given knowledge can not be relied on to answer the question. This strategy is helpful since the retrieved knowledge might be noisy. We use a RoBERTa-large (Liu *et al.*, 2019b) as the text encoder, whose inputs are {[SEP] question [SEP] [“unanswerable”], caption, knowledge [SEP]}. Then each token representation is fed to two linear layers: one predicts a score for a token being the start token, and the other predicts a score for the end token. We apply the softmax function to get the probability of each token being a start and end token. The training objective is to maximize the probability of the ground truth start and end token.

11.2 Experiments and Results

We use a state-of-the-art vision-language model, LXMERT (Tan and Bansal, 2019a), as the baselines and apply Captioning and Optical Character Recognition (OCR) results to the OK-VQA dataset to the original LXMERT model.

LXMERT LXMERT is a BERT-based cross-modality model pretrained on five different VQA datasets: MS COCO (Lin *et al.*, 2014), Visual Genome (Krishna *et al.*, 2017), VQA v2.0 (Antol *et al.*, 2015), GQA balanced version (Hudson and Manning, 2019) and VG-QA (Zhu *et al.*, 2016). We fine-tune LXMERT on OK-VQA and surprisingly find that LXMERT ranks higher than most of the SOTA models, for which reason we set LXMERT as our baseline model.

LXMERT with OCR The OCR technique captures the textual contents from the image and transfers them into characters. Here we use Google Vision API¹ to extract the texts from images. After the noise deduction step filtering all non-English words, we attach the OCR results after the question and then sent them into the LXMERT model. Our experiment shows that the OCR result helps to address the OK-VQA task.

LXMERT with Captioning Similar to OCR, we also experiment with adding captioning when training the LXMERT model. The captions are generated by the advanced model Oscar Li *et al.* (2020b) and attached to each question when sent into the LXMERT model. Our result shows that captioning improves the performance of the LXMERT model, and therefore, we put the LXMERT with captioning as a baseline as well.

Result Table 11.1 shows that our best model based on Caption-DPR and EReader outperforms previous methods and establishes the new state-of-the-art result on the OK-VQA challenge. Interestingly, the LXMERT baseline without utilizing any knowledge achieves better performance than KRISP (Marino *et al.*, 2020) and Concept-Bert (Gardères *et al.*, 2020) which leverage external knowledge. Incorporating OCR

¹<https://cloud.google.com/vision/>

and captioning further improve the baseline accuracy by 1% and 1.6%, respectively.

11.3 Discussion and Summary

We introduce the first extractive reader designed for the Visual Question Answering (VQA) task, capable of extracting answer spans from provided knowledge sources. In contrast to conventional classification-based models that rely on predetermined answer lists derived from training data, our extractive VQA reader demonstrates improved generalization for answer spans not seen during training. Additionally, we highlight the significant influence of retrieved knowledge quality on downstream VQA tasks, indicating the importance of developing more precise multimodal retrieval models.

Method	Knowledge Src.	Acc.	Open Acc.
Existing Method			
KRISP (Marino <i>et al.</i> , 2020)	W & C	32.3	-
ConceptBert (Gardères <i>et al.</i> , 2020)	C	33.7	-
MAVEEx (Wu <i>et al.</i> , 2021)	W & C & GI	38.7	-
Baselines			
LXMERT (without pretraining)	-	18.9	25.5
LXMERT	-	36.2	42.6
LXMERT + OCR	-	37.2	42.2
LXMERT + Caption	-	37.8	45.6
LXMERT + OCR + Caption	-	37.2	44.5
Visual Retriever-Reader			
BM25 + CReader	GS	35.13	43.8
BM25 + EReader	GS	32.10	40.6
Image-DPR + CReader	GS	34.64	43.2
Image-DPR + EReader	GS	33.95	41.7
Caption-DPR + CReader	GS	36.78	43.4
Caption-DPR + EReader	GS	39.20	47.3
Caption-DPR + EReader [†]	GS	59.22	66.6

Table 11.1: Performance on the OK-VQA Test-split. Our model outperforms existing methods. [†] means given oracle knowledge to the reader. GS-Google Search (Training Corpus). W-Wikipedia, C-ConceptNet, GI-Google Image, Acc-Accuracy.

Chapter 12

EVALUATION FOR VQA

One important style of visual question answering (VQA) task involves open-ended responses such as free-form answers or fill-in-the-blanks. The possibility of multiple correct answers and multi-word responses makes the evaluation of open-ended tasks harder, which has forced VQA datasets to restrict answers to be a single word or a short phrase. Despite enforcing these constraints, from our analysis of the GQA dataset (Hudson and Manning, 2019), we noticed that a significant portion of the visual questions have issues. For example, a question “*Who is holding the bat?*” has only one ground truth answer “*batter*” while other reasonable answers like “*batsman*”, “*hitter*” are not credited. We identified six different types of issues with the dataset and illustrated them in Table 12.1. A large-scale human-study conducted by (Gurari and Grauman, 2017) on VQA (Antol *et al.*, 2015) and VizWiz (Gurari *et al.*, 2019) found that almost 50% questions in these datasets have multiple possible answers. datasets had similar observations. The above evidence suggests that it is unfair to penalize models if their predicted answer is correct in a given context but does not match the ground truth answer.

12.1 Existing Evaluation

For open-ended VQA tasks, the standard accuracy metric can be too stringent as it requires a predicted answer to exactly match the ground-truth answer. To deal with different interpretations of words and multiple correct answers, Malinowski and Fritz (2014) defined a WUPS scoring from lexical databases with Wu-Palmer similarity (Wu and Palmer, 1994). Abdelkarim *et al.* (2020) proposed a soft matching metric based

Issue Type	Definition	%
[1] Synonym and Hypernym	Synonym or hypernym of the ground-truth can also be considered as a correct answer for a given question-image pair.	9.1
[2] Singular/Plural	Singular or plural of the ground-truth can also be considered as a correct answer for a given question-image pair.	1.0
[3] Ambiguous Objects	Question refers to an object but the image contains multiple such objects that can lead to different possible answers.	5.8
[4] Multiple Correct Answers	If a given image-question pair is not precise, annotators might have different opinion which leads to multiple correct answers	7.0
[5] Missing Object(s)	Object referred in the question is not clearly visible in image.	4.3
[6] Wrong Label	The ground-truth answer to a question-image pair is incorrect.	6.7

Table 12.1: Six types of issues observed in the GQA dataset, their definition and their distribution observed in manual review of 600 samples from testdev balanced split.

on wordNet (Miller, 1998) and word2vec (Mikolov *et al.*, 2013). Different from them, we incorporate more advanced NLP resources tools to generate answer sets and rely on textual entailment to validate semantics for robustness. Semantic evaluation has also discussed for other tasks, such as image captioning generation (Feinglass and Yang, 2021).

12.2 Semantic Evaluation: Alternative Answer Set

To credit answers with semantically close meaning as the ground-truth, we propose a workflow that can be visualized from Figure 12.1. Each item in VQA dataset consists of $\langle I, Q, GT \rangle$, where I is an image, Q is a question, and GT is a ground-truth answer. We define an Alternative Answer Set (AAS) as a collection of phrases $\{A_1, A_2, A_3, \dots, A_n\}$ such that A_i replaced with GT is still a valid answer to the given Image-Question pair. We construct AAS for each unique ground-truth automatically from two knowledge bases: Wordnet (Miller, 1998) and ConceptNet (Liu and Singh,

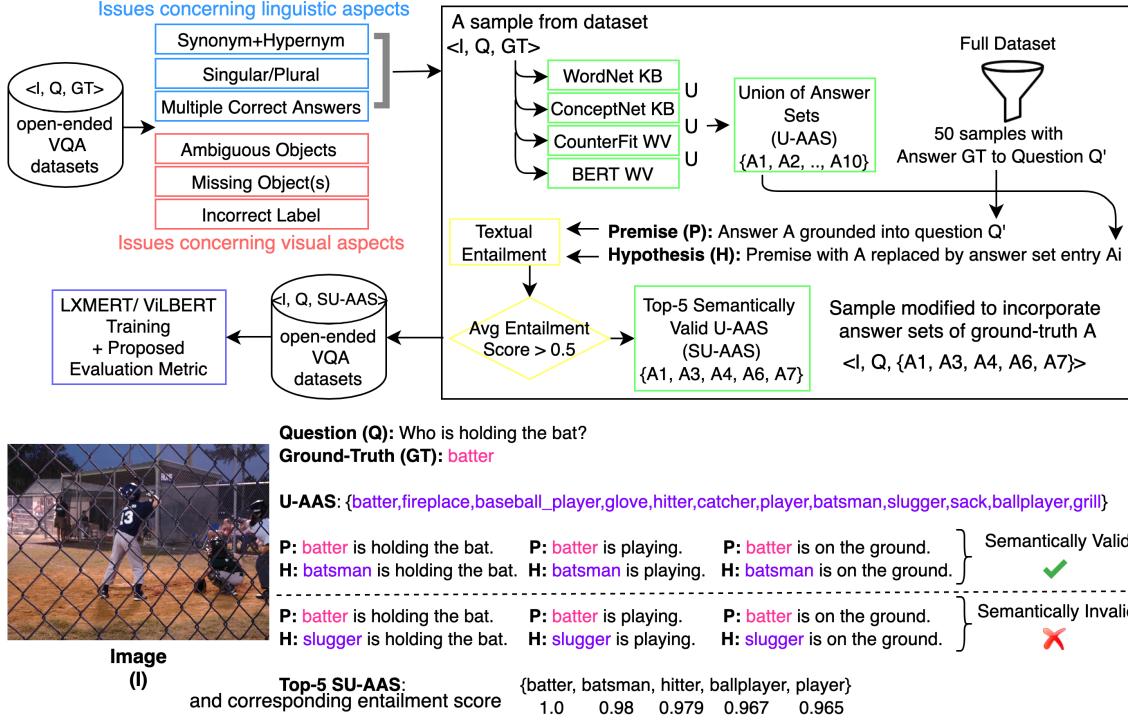


Figure 12.1: (top) The workflow for generating Alternative Answer Set (AAS) for VQA datasets (bottom) An example from GQA dataset showing semantically valid AAS for the answer ‘batter’ generated using above workflow

2004), two word embeddings: BERT (Devlin *et al.*, 2018) and counter-fitting (Mrkšić *et al.*, 2016). We assign a semantic score to each alternative answer by textual entailment and introduce the AAS metric.

Semantic Union AAS We take a union of four methods to find all alternative answers. For example, “stuffed animal” is semantic similar to “teddy bear”, which appears in the AAS based on BERT but not in WordNet. However, the union might include phrases that we want to distinguish from the label like “man” is in the AAS of “woman” when using the BERT-based approach. For this reason, we employ the textual entailment technique to compute a semantic score of each alternative answer. For each label, we first obtain 50 sentences containing the ground-truth label from GQA dataset. We take each sentence as a premise, replace the label in this sentence

with a phrase in its AAS as a hypothesis to generate an entailment score between 0-1. Specifically, we use publicly available RoBERTa (Liu *et al.*, 2019b) model trained on SNLI (Stanford Natural Language Inference) (Bowman *et al.*, 2015b) dataset for entailment computation. The semantic score of the alternative answer is the average of 50 entailment scores. If the semantic score is lower than the threshold of 0.5, then this alternative answer is thrown out. We choose 0.5 since it is the middle of 0 and 1.

Lastly, we sort the AAS by semantic score and keep the top K in the semantic union AAS, annotated by SU-AAS. We experiment with different values of K from 2 to 10, and decide K to be 6, a trade-off between accuracy and robustness. Note that the performance of textual entailment model is a contributing factor in obtaining quality AAS. Therefore, we recommend using the state-of-the-art entailment model when our proposed method is applied on other VQA datasets.

Evaluation Metric Based on AAS We propose AAS metric and semantic score: given a question Q_i , an image I_i , the alternative answer set of GT_i denoted by S_{GT_i} , the prediction of model P_i is correct if and only if it is found in S_{GT_i} , and the score of P_i is $S_{GT_i}(P_i)$, where $S_{GT_i}(P_i)$ is the semantic score of P_i . Mathematically,

$$Acc(Q_i, I_i, S_{GT_i}, P_i) = \begin{cases} S_{GT_i}(P_i) & \text{if } P_i \in S_{GT_i} \\ 0 & \text{else} \end{cases}$$

12.3 Experiments and Results

In this section, we first show that the performance of vision-language models on two datasets is improved based on the AAS metric. Then, we describe our experiment to incorporate AAS with one model on GQA dataset. Last, we verify the correctness of AAS by human evaluation.

12.3.1 Baseline Methods

We select two top Vision-and-Language models, ViLBERT (Lu *et al.*, 2019) and LXMERT (Tan and Bansal, 2019b) and evaluate their performances based on the AAS metric. From Table 12.2, we see that for the GQA dataset, LXMERT and ViLBERT have 4.49%, 4.26% improvements on union AAS metric separately. For VQA2.0 dataset, LXMERT and ViLBERT have 0.82%, 0.53% improvements on union AAS metric separately. It is expected that the improvement on VQA2.0 dataset is less than GQA since the former dataset already provides multiple correct answers. Figure 12.2 shows the impacts of the value K of Union AAS on the scores. From the figure, we see that when K increases from 2 to 6, the score gets increased significantly, and slightly when k increases from 6 to 9, but not increases more after K is 9. Since values 7 and 8 do not significantly improve the score, and the value 9 introduces noise, we take the top 6 as the SU-AAS.

Dataset	Model	Original Metric	WordNet	BERT	CounterFit	ConceptNet	Union
GQA	LXMERT	60.06	61.79	62.69	62.75	63.58	64.55
(testdev)	ViLBERT	60.13	61.90	62.69	62.74	63.67	64.39
VQA	LXMERT	69.98	70.21	70.54	70.33	70.52	70.80
(valid)	ViLBERT	77.65	77.82	78.10	77.93	78.06	78.28

Table 12.2: The evaluation of two models on GQA and VQA with original metric and AAS based metrics.

12.3.2 Training with AAS

We incorporate SU-AAS of ground truth in training phase, so the model learns that more than one answer for a given example can be correct. We train LXMERT on GQA dataset with this objective.

Table 12.3 shows the results of LXMERT trained with AAS compared with the baseline. Not surprisingly, the performance evaluated on the original method drops

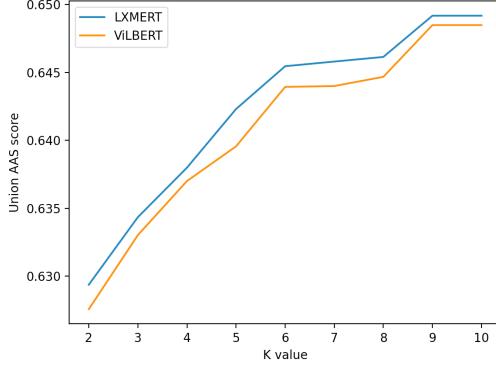


Figure 12.2: Union AAS score of different value of K

because the model has a higher chance to predict answers in AAS, which are different from the ground truth, and thus the performance evaluated on SU-AAS metric increases.

Dataset	Exact Matching Accuracy		SU-AAS Accuracy	
	LXMERT	$LXMERT_{AAS}$	LXMERT	$LXMERT_{AAS}$
GQA(testdev)	60.06	59.02	64.55	65.22

Table 12.3: Incorporate AAS in the training phase of LXMERT ($LXMERT_{AAS}$) on GQA dataset.

12.3.3 Evaluation of AAS

To validate the correctness of AAS, we measure the correlation between human judgment and AAS. Specifically, for each label of GQA, we take the SU-AAS and ask three annotators to justify if alternative answers in AAS can replace the label. If the majority of annotators agree upon, we keep the answer in the AAS, remove otherwise. In this way, we collect the human-annotated AAS. We compare the human-annotated AAS with each automatically generated AAS. We take the intersection over union (IoU) score to evaluate the correlation between automatic approach and human annotation: a higher IoU score means stronger alignment.

Method	WordNet	BERT	CounterFit	ConceptNet	Union
IoU%	48.25	56.18	58.95	58.39	80.5

Table 12.4: The IoU scores between human annotations and AAS based on five approaches.

12.4 Discussion and Summary

To evaluate a model from a semantic point of view, we define an alternative answer set (AAS). We develop a workflow to automatically create robust AAS for ground truth answers in the dataset using Textual Entailment. Additionally, we did human verification to assess the quality of automatically generated AAS. The high agreement score indicates that entailment model is doing a careful job of filtering relevant answers. From experiments on two models and two VQA datasets, we show the effectiveness of AAS-based evaluation using our proposed metric.

AAS can be applied to other tasks, for example, machine translation. BLEU(Papineni *et al.*, 2002) score used to evaluate machine translation models incorporates an average of n-gram precision but does not consider the synonymy. Therefore, METEOR (Banerjee and Lavie, 2005) was proposed to overcome this problem. However, METEOR only relies on the synset of WordNet to get the synonyms. Our proposed AAS has the advantage of both knowledge base and word embeddings, which would help better evaluate translation tasks.

Chapter 13

CONCLUSION

Motivated by the importance of information retrieval and question answering tasks in the modern digital landscape, where multimodal data such as text and images are ubiquitous, this thesis strives to advance these two essential AI tasks in both textual and multimodal domains. Our goal is to create more dependable real-world applications that contribute positively to various aspects of human experiences.

To construct accurate, efficient, and robust information retrieval and question answering systems, we have devised innovative model architectures(§2, §3, §5, §6, §9, §11), pretraining tasks (§2, §6), and data augmentation techniques (§4) to enhance existing systems and methods. We have also conducted a thorough analysis across various existing models to understand their strengths and weaknesses in addressing domain shift challenges encountered in real-life situations(§7, §8). Moreover, we introduced new benchmarks (§3,§6, §10) and evaluation metric (§12) to more effectively assess the robustness and reasoning capacity of both information retrieval and question answering systems. Although our proposed system has achieved significant improvements over existing methods, there is still room for further research and refinement. We will discuss future research directions in the subsequent section.

13.1 Future Work

My thesis has concentrated on two essential AI challenges: information retrieval and question answering. In 2023, Generative AI has taken center stage, as demonstrated by the rise of ChatGPT, which has led to numerous individuals benefiting from AI technologies and being enthusiastic about the capabilities of generative models.

Motivated by AI's immense potential to enhance and revolutionize human life, my future research will explore various avenues to amplify the advantages of AI, ultimately contributing to the betterment of human existence.

13.1.1 A Biomedical Dialogue Agent Using IR and QA

My first focus will be on the intersection of AI and the biomedical and healthcare fields. Although generative AI has demonstrated remarkable performance in general domains, significant gaps remain in biomedical areas that pose distinct challenges, such as privacy concerns, limited training data, and a heavy reliance on domain-specific knowledge. Notable advancements have been made in the biomedical field, including multi-task learning and domain-specific biomedical language models. However, the interpretability of these techniques remains limited, hindering their real-world applications. Developing explainable biomedical models is crucial for enhancing the trustworthiness of these models, which in turn, facilitates their implementation in real-life situations.

13.1.2 Retrieval-augmented LM to Address Hallucination

While LLMs have shown impressive capability of performing well on general tasks, they still suffer from many issues, one of them is hallucination. A combination of retrievers and LLMs is a promising approach to address the hallucination. LLMs should know when to retrieve external knowledge and when it can rely on its internal knowledge. Like human, for some knowledge that we are very familiar with, we do not need to search, however, for some other unfamiliar or uncertain domains, we tend to gather solid evidences before making conclusion. LLMs should also have such dynamic searching ability to accommodate different situations.

13.1.3 Efficient and Small Language Model

The scaling law shows that models' capacity expands with the growth of parameters, and for some exciting capacities, such as in-context learning, only appear in large large models. However, due to computational power limitations, only few parties, such as major industry corporations, can afford the exceptionally costly training processes.

Smaller language models are more cost-effective to train, which raises a significant research question: Are a large number of parameters essential for general intelligence? Considering that our human brain contains 86 billion of neurons interconnected, it is possible that intelligent models might also require a substantial number of parameters.

Nonetheless, looking back at history, such as the development of computers, the first computer emerged in 1945 and occupied an entire room, whereas now, every cell phone is a computer that is far more capable than the room-sized predecessor. Language models may follow a similar trajectory, eventually leading to personal, portable language models. Pursuing efficient and compact language models is an inevitable step towards this objective.

REFERENCES

- Abdelkarim, S., P. Achlioptas, J. Huang, B. Li, K. Church and M. Elhoseiny, “Long-tail visual relationship recognition with a visiolinguistic hubless loss”, arXiv preprint arXiv:2004.00436 (2020).
- Ahmad, A., N. Constant, Y. Yang and D. M. Cer, “Reqa: An evaluation for end-to-end answer retrieval models”, ArXiv **abs/1907.04780**, 137–146 (2019).
- Alayrac, J.-B., J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, “Flamingo: a visual language model for few-shot learning”, arXiv preprint arXiv:2204.14198 (2022).
- Alberti, C., D. Andor, E. Pitler, J. Devlin and M. Collins, “Synthetic QA corpora generation with roundtrip consistency”, in “Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics”, pp. 6168–6173 (Association for Computational Linguistics, Florence, Italy, 2019), URL <https://www.aclweb.org/anthology/P19-1620>.
- Almeida, T. and S. Matos, “Bit. ua at bioasq 8: Lightweight neural document ranking with zero-shot snippet retrieval.”, in “CLEF (Working Notes)”, (2020).
- Alzantot, M., Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava and K.-W. Chang, “Generating natural language adversarial examples”, in “Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing”, pp. 2890–2896 (Association for Computational Linguistics, Brussels, Belgium, 2018), URL <https://www.aclweb.org/anthology/D18-1316>.
- Antol, S., A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick and D. Parikh, “VQA: visual question answering”, in “2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7–13, 2015”, pp. 2425–2433 (IEEE Computer Society, 2015), URL <https://doi.org/10.1109/ICCV.2015.279>.
- Arabzadeh, N., X. Yan and C. L. Clarke, “Predicting efficiency/effectiveness trade-offs for dense vs. sparse retrieval strategy selection”, in “Proceedings of the 30th ACM International Conference on Information & Knowledge Management”, pp. 2862–2866 (2021).
- Arjovsky, M., L. Bottou, I. Gulrajani and D. Lopez-Paz, “Invariant risk minimization”, arXiv preprint arXiv:1907.02893 (2019).
- Banerjee, P. and C. Baral, “Knowledge fusion and semantic knowledge ranking for open domain question answering”, ArXiv **abs/2004.03101** (2020).
- Banerjee, P., C. Baral, M. Luo, A. Mitra, K. Pal, T. C. Son and N. Varshney, “Can transformers reason about effects of actions?”, arXiv:2012.09938 (2020).

- Banerjee, P., T. Gokhale and C. Baral, “Self-supervised test-time learning for reading comprehension”, in “Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies”, pp. 1200–1211 (2021a).
- Banerjee, P., T. Gokhale, Y. Yang and C. Baral, “Weaqa: Weak supervision via captions for visual question answering”, in “Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021”, pp. 3420–3435 (2021b).
- Banerjee, P., T. Gokhale, Y. Yang and C. Baral, “WeaQA: Weak supervision via captions for visual question answering”, in “Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021”, pp. 3420–3435 (Association for Computational Linguistics, Online, 2021c), URL <https://aclanthology.org/2021.findings-acl.302>.
- Banerjee, S. and A. Lavie, “Meteor: An automatic metric for mt evaluation with improved correlation with human judgments”, in “Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization”, pp. 65–72 (2005).
- Beery, S., G. Van Horn and P. Perona, “Recognition in terra incognita”, in “Proceedings of the European conference on computer vision (ECCV)”, pp. 456–473 (2018).
- Bhargava, P., A. Drozd and A. Rogers, “Generalization in NLI: Ways (not) to go beyond simple heuristics”, in “Proceedings of the Second Workshop on Insights from Negative Results in NLP”, pp. 125–135 (Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021), URL <https://aclanthology.org/2021.insights-1.18>.
- Bilotti, M. W., P. Ogilvie, J. Callan and E. Nyberg, “Structured retrieval for question answering”, in “SIGIR”, (2007).
- Bowman, S. R., G. Angeli, C. Potts and C. D. Manning, “A large annotated corpus for learning natural language inference”, arXiv preprint arXiv:1508.05326 (2015a).
- Bowman, S. R., G. Angeli, C. Potts and C. D. Manning, “A large annotated corpus for learning natural language inference”, in “Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing”, pp. 632–642 (Association for Computational Linguistics, Lisbon, Portugal, 2015b), URL <https://www.aclweb.org/anthology/D15-1075>.
- Bras, R. L., S. Swayamdipta, C. Bhagavatula, R. Zellers, M. E. Peters, A. Sabharwal and Y. Choi, “Adversarial filters of dataset biases”, in “Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event”, vol. 119 of *Proceedings of Machine Learning Research*, pp. 1078–1088 (PMLR, 2020), URL <http://proceedings.mlr.press/v119/bras20a.html>.
- Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners”, Advances in neural information processing systems **33**, 1877–1901 (2020).

- Bruch, S., C. Lucchese and F. M. Nardini, “Reneuir: Reaching efficiency in neural information retrieval”, in “Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval”, pp. 3462–3465 (2022).
- Cakaloglu, T., C. Szegedy and X. Xu, “Text embeddings for retrieval from a large knowledge base”, in “International Conference on Research Challenges in Information Science”, (Springer, 2020).
- Chai, L., J.-Y. Zhu, E. Shechtman, P. Isola and R. Zhang, “Ensembling with deep generative views”, in “Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition”, pp. 14997–15007 (2021).
- Chang, W.-C., F. Yu, Y.-W. Chang, Y. Yang and S. Kumar, “Pre-training tasks for embedding-based large-scale retrieval”, ArXiv [abs/2002.03932](https://arxiv.org/abs/2002.03932) (2020).
- Chang, Y., M. Narang, H. Suzuki, G. Cao, J. Gao and Y. Bisk, “Webqa: Multihop and multimodal qa”, arXiv preprint arXiv:2109.00590 (2021).
- Chen, D., *Neural reading comprehension and beyond* (Stanford University, 2018).
- Chen, D., A. Fisch, J. Weston and A. Bordes, “Reading Wikipedia to answer open-domain questions”, in “Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)”, pp. 1870–1879 (Association for Computational Linguistics, Vancouver, Canada, 2017a), URL <https://aclanthology.org/P17-1171>.
- Chen, D., A. Fisch, J. Weston and A. Bordes, “Reading Wikipedia to answer open-domain questions”, in “Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)”, pp. 1870–1879 (Association for Computational Linguistics, Vancouver, Canada, 2017b), URL <https://aclanthology.org/P17-1171>.
- Chen, T. and B. Van Durme, “Discriminative information retrieval for question answering sentence selection”, in “EACL”, (ACL, Valencia, Spain, 2017), URL <https://aclanthology.org/E17-2114>.
- Chen, T., M. Zhang, J. Lu, M. Bendersky and M. Najork, “Out-of-domain semantics to the rescue! zero-shot hybrid retrieval models”, in “European Conference on Information Retrieval”, pp. 95–110 (Springer, 2022a).
- Chen, X., H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár and C. L. Zitnick, “Microsoft coco captions: Data collection and evaluation server”, arXiv preprint arXiv:1504.00325 (2015).
- Chen, X., K. Lakhotia, B. Oğuz, A. Gupta, P. Lewis, S. Peshterliev, Y. Mehdad, S. Gupta and W.-t. Yih, “Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one?”, arXiv preprint arXiv:2110.06918 (2021).

- Chen, X., J. Luo, B. He, L. Sun and Y. Sun, “Towards robust dense retrieval via local ranking alignment”, in “Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI”, pp. 1980–1986 (2022b).
- Choi, E., H. He, M. Iyyer, M. Yatskar, W.-t. Yih, Y. Choi, P. Liang and L. Zettlemoyer, “QuAC: Question answering in context”, in “Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing”, pp. 2174–2184 (Association for Computational Linguistics, Brussels, Belgium, 2018), URL <https://aclanthology.org/D18-1241>.
- Clark, C., M. Yatskar and L. Zettlemoyer, “Don’t take the easy way out: Ensemble based methods for avoiding known dataset biases”, in “Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)”, pp. 4069–4082 (Association for Computational Linguistics, Hong Kong, China, 2019), URL <https://www.aclweb.org/anthology/D19-1418>.
- Clark, K., M. Luong, Q. V. Le and C. D. Manning, “ELECTRA: pre-training text encoders as discriminators rather than generators”, in “8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020”, (OpenReview.net, 2020a), URL <https://openreview.net/forum?id=r1xMH1BtvB>.
- Clark, P., O. Tafjord and K. Richardson, “Transformers as soft reasoners over language”, in “IJCAI”, edited by C. Bessiere (2020b), URL <https://doi.org/10.24963/ijcai.2020/537>.
- Cohen, D., L. Yang and W. Croft, “Wikipediaqa: A benchmark collection for research on non-factoid answer passage retrieval”, The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval **abs/1805.03797** (2018).
- Conneau, A. and D. Kiela, “Senteval: An evaluation toolkit for universal sentence representations”, ArXiv **abs/1803.05449** (2018).
- Cormack, G. V., C. L. Clarke and S. Buettcher, “Reciprocal rank fusion outperforms condorcet and individual rank learning methods”, in “Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval”, pp. 758–759 (2009).
- Dai, Z. and J. 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”, pp. 1533–1536 (2020).
- Dai, Z., C. Xiong, J. Callan and Z. Liu, “Convolutional neural networks for soft-matching n-grams in ad-hoc search”, WSDM (2018).
- Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding”, arXiv preprint arXiv:1810.04805 (2018).

Devlin, J., M.-W. Chang, K. Lee and K. 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)”, pp. 4171–4186 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019a), URL <https://aclanthology.org/N19-1423>.

Devlin, J., M.-W. Chang, K. Lee and K. 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)”, pp. 4171–4186 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019b), URL <https://www.aclweb.org/anthology/N19-1423>.

Dhingra, B., K. Mazaitis and W. W. Cohen, “Quasar: Datasets for question answering by search and reading”, ArXiv preprint **abs/1707.03904**, URL <https://arxiv.org/abs/1707.03904> (2017).

Dong, L., N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou and H. Hon, “Unified language model pre-training for natural language understanding and generation”, in “Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada”, edited by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox and R. Garnett, pp. 13042–13054 (2019), URL <https://proceedings.neurips.cc/paper/2019/hash/c20bb2d9a50d5ac1f713f8b34d9aac5a-Abstract.html>.

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale”, arXiv preprint arXiv:2010.11929 (2020).

Du, X. and C. Cardie, “Harvesting paragraph-level question-answer pairs from Wikipedia”, in “Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)”, (Association for Computational Linguistics, Melbourne, Australia, 2018), URL <https://www.aclweb.org/anthology/P18-1177>.

Dunn, M., L. Sagun, M. Higgins, V. U. Güney, V. Cirik and K. Cho, “Searchqa: A new q&a dataset augmented with context from a search engine”, ArXiv **abs/1704.05179** (2017).

Fabbri, A., P. Ng, Z. Wang, R. Nallapati and B. Xiang, “Template-based question generation from retrieved sentences for improved unsupervised question answering”, in “Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics”, pp. 4508–4513 (Association for Computational Linguistics, Online, 2020), URL <https://www.aclweb.org/anthology/2020.acl-main.413>.

Faghri, F., D. J. Fleet, J. R. Kiros and S. Fidler, “Vse++: Improving visual-semantic embeddings with hard negatives”, arXiv preprint arXiv:1707.05612 (2017).

Fang, Y., S. Sun, Z. Gan, R. Pillai, S. Wang and J. Liu, “Hierarchical graph network for multi-hop question answering”, in “Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)”, pp. 8823–8838 (Association for Computational Linguistics, Online, 2020), URL <https://aclanthology.org/2020.emnlp-main.710>.

Feinglass, J. and Y. Yang, “SMURF: SeMantic and linguistic UndeRstanding fusion for caption evaluation via typicality analysis”, in “Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)”, pp. 2250–2260 (Association for Computational Linguistics, Online, 2021), URL <https://aclanthology.org/2021.acl-long.175>.

Fisch, A., A. Talmor, R. Jia, M. Seo, E. Choi and D. Chen, “MRQA 2019 shared task: Evaluating generalization in reading comprehension”, in “Proceedings of 2nd Machine Reading for Reading Comprehension (MRQA) Workshop at EMNLP”, (2019).

Galárraga, L., C. Teflioudi, K. Hose and F. M. Suchanek, “Fast rule mining in ontological knowledge bases with amie++”, *The VLDB Journal* **24**, 6, 707–730 (2015).

Ganin, Y., E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand and V. Lempitsky, “Domain-adversarial training of neural networks”, *The journal of machine learning research* **17**, 1, 2096–2030 (2016).

Gao, F., Q. Ping, G. Thattai, A. Reganti, Y. N. Wu and P. Natarajan, “A thousand words are worth more than a picture: Natural language-centric outside-knowledge visual question answering”, arXiv preprint arXiv:2201.05299 (2022).

Gao, L. and J. Callan, “Unsupervised corpus aware language model pre-training for dense passage retrieval”, in “Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)”, pp. 2843–2853 (Association for Computational Linguistics, Dublin, Ireland, 2022), URL <https://aclanthology.org/2022.acl-long.203>.

Gardères, F., M. Ziaeefard, B. Abeloos and F. Lecue, “ConceptBert: Concept-aware representation for visual question answering”, in “Findings of the Association for Computational Linguistics: EMNLP 2020”, pp. 489–498 (Association for Computational Linguistics, Online, 2020), URL <https://aclanthology.org/2020.findings-emnlp.44>.

Gokhale, T., P. Banerjee, C. Baral and Y. Yang, “MUTANT: A training paradigm for out-of-distribution generalization in visual question answering”, in “Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)”, pp. 878–892 (Association for Computational Linguistics, Online, 2020a), URL <https://aclanthology.org/2020.emnlp-main.63>.

- Gokhale, T., P. Banerjee, C. Baral and Y. Yang, “Vqa-lol: Visual question answering under the lens of logic”, in “European conference on computer vision”, pp. 379–396 (Springer, 2020b).
- Gokhale, T., S. Mishra, M. Luo, B. S. Sachdeva and C. Baral, “Generalized but not robust? comparing the effects of data modification methods on out-of-domain generalization and adversarial robustness”, arXiv preprint arXiv:2203.07653 (2022).
- Goodfellow, I. J., J. Shlens and C. Szegedy, “Explaining and harnessing adversarial examples”, in “3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings”, edited by Y. Bengio and Y. LeCun (2015), URL <http://arxiv.org/abs/1412.6572>.
- Groeneveld, D., T. Khot, Mausam and A. Sabharwal, “A simple yet strong pipeline for HotpotQA”, in “Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)”, pp. 8839–8845 (Association for Computational Linguistics, Online, 2020), URL <https://aclanthology.org/2020.emnlp-main.711>.
- Gui, L., B. Wang, Q. Huang, A. Hauptmann, Y. Bisk and J. Gao, “Kat: A knowledge augmented transformer for vision-and-language”, arXiv preprint arXiv:2112.08614 (2021).
- Gulrajani, I. and D. Lopez-Paz, “In search of lost domain generalization”, in “International Conference on Learning Representations”, (2020).
- Guo, J., Y. Fan, Q. Ai and W. Croft, “A deep relevance matching model for ad-hoc retrieval”, CIKM (2016).
- Guo, M., Y. Yang, D. Cer, Q. Shen and N. Constant, “MultiReQA: A cross-domain evaluation for Retrieval question answering models”, in “Proceedings of the Second Workshop on Domain Adaptation for NLP”, pp. 94–104 (Association for Computational Linguistics, Kyiv, Ukraine, 2021), URL <https://aclanthology.org/2021.adaptnlp-1.10>.
- Gurari, D. and K. Grauman, “Crowdverge: Predicting if people will agree on the answer to a visual question”, in “Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems”, pp. 3511–3522 (2017).
- Gurari, D., Q. Li, C. Lin, Y. Zhao, A. Guo, A. Stangl and J. P. Bigham, “Vizwiz-priv: A dataset for recognizing the presence and purpose of private visual information in images taken by blind people”, in “IEEE CVPR”, pp. 939–948 (2019).
- Guu, K., K. Lee, Z. Tung, P. Pasupat and M. Chang, “Retrieval augmented language model pre-training”, in “International Conference on Machine Learning”, pp. 3929–3938 (PMLR, 2020a).
- Guu, K., K. Lee, Z. Tung, P. Pasupat and M.-W. Chang, “Realm: Retrieval-augmented language model pre-training”, ArXiv **abs/2002.08909** (2020b).
- Guu, K., K. Lee, Z. Tung, P. Pasupat and M.-W. Chang, “Realm: Retrieval-augmented language model pre-training”, ArXiv **abs/2002.08909** (2020c).

- Habernal, I., H. Wachsmuth, I. Gurevych and B. Stein, “The argument reasoning comprehension task: Identification and reconstruction of implicit warrants”, in “NAACL-HLT”, (2018), URL <https://aclanthology.org/N18-1175>.
- Hannun, A., C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, “Deep speech: Scaling up end-to-end speech recognition”, arXiv preprint arXiv:1412.5567 (2014).
- Harwood, B., V. Kumar BG, G. Carneiro, I. Reid and T. Drummond, “Smart mining for deep metric learning”, in “Proceedings of the IEEE International Conference on Computer Vision”, pp. 2821–2829 (2017).
- Hendrycks, D. and T. G. Dietterich, “Benchmarking neural network robustness to common corruptions and perturbations”, 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=HJz6tiCqYm>.
- Hendrycks, D., X. Liu, E. Wallace, A. Dziedzic, R. Krishnan and D. Song, “Pretrained transformers improve out-of-distribution robustness”, in “Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics”, pp. 2744–2751 (2020a).
- Hendrycks, D., N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer and B. Lakshminarayanan, “Augmix: A simple data processing method to improve robustness and uncertainty”, in “8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020”, (OpenReview.net, 2020b), URL <https://openreview.net/forum?id=S1gmrxFHvB>.
- Hochreiter, S. and J. Schmidhuber, “Long short-term memory”, Neural computation **9**, 8, 1735–1780 (1997).
- Hofstätter, S., S.-C. Lin, J.-H. Yang, J. Lin and A. Hanbury, “Efficiently teaching an effective dense retriever with balanced topic aware sampling”, in “Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval”, pp. 113–122 (2021).
- Hosking, T. and M. Lapata, “Factorising meaning and form for intent-preserving paraphrasing”, 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)”, (Association for Computational Linguistics, Online, 2021), URL <https://aclanthology.org/2021.acl-long.112>.
- Hudson, D. A. and C. D. Manning, “GQA: A new dataset for real-world visual reasoning and compositional question answering”, in “IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019”, pp. 6700–6709 (Computer Vision Foundation / IEEE, 2019), URL http://openaccess.thecvf.com/content_CVPR_2019/html/Hudson_GQA_A_New_Dataset_for_Real-World_Visual_Reasoning_and_Compositional_CVPR_2019_paper.html.

- Hui, K., A. Yates, K. Berberich and G. de Melo, “PACRR: A position-aware neural IR model for relevance matching”, in “EMNLP”, (ACL, Copenhagen, Denmark, 2017), URL <https://aclanthology.org/D17-1110>.
- Humeau, S., K. Shuster, M.-A. Lachaux and J. Weston, “Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring”, in “ICLR”, (2020).
- Izacard, G. and E. 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”, pp. 874–880 (Association for Computational Linguistics, Online, 2021), URL <https://aclanthology.org/2021.eacl-main.74>.
- Jain, A., M. Kothyari, V. Kumar, P. Jyothi, G. Ramakrishnan and S. Chakrabarti, “Select, substitute, search: A new benchmark for knowledge-augmented visual question answering”, ArXiv preprint **abs/2103.05568**, URL <https://arxiv.org/abs/2103.05568> (2021).
- Jegou, H., M. Douze and C. Schmid, “Product quantization for nearest neighbor search”, IEEE transactions on pattern analysis and machine intelligence **33**, 1, 117–128 (2010).
- Joshi, M., E. Choi, D. Weld and L. 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)”, pp. 1601–1611 (Association for Computational Linguistics, Vancouver, Canada, 2017), URL <https://www.aclweb.org/anthology/P17-1147>.
- Joshi, P., S. Aditya, A. Sathe and M. Choudhury, “TaxiNLI: Taking a ride up the NLU hill”, in “CoNLL”, (2020), URL <https://aclanthology.org/2020.conll-1.4>.
- Karpukhin, V., B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen and W.-t. Yih, “Dense passage retrieval for open-domain question answering”, in “Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)”, pp. 6769–6781 (Association for Computational Linguistics, Online, 2020a), URL <https://aclanthology.org/2020.emnlp-main.550>.
- Karpukhin, V., B. Oğuz, S. Min, P. Lewis, L. Y. Wu, S. Edunov, D. Chen and W. tau Yih, “Dense passage retrieval for open-domain question answering”, in “EMNLP”, (2020b).
- Kassner, N. and H. Schütze, “Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly”, in “Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics”, pp. 7811–7818 (Association for Computational Linguistics, Online, 2020), URL <https://www.aclweb.org/anthology/2020.acl-main.698>.
- Kazaryan, A., U. Sazanovich and V. Belyaev, “Transformer-based open domain biomedical question answering at bioasq8 challenge.”, in “CLEF (Working Notes)”, (2020).

Khattab, O., C. Potts and M. Zaharia, “Relevance-guided supervision for openqa with colbert”, *Transactions of the Association for Computational Linguistics* **9**, 929–944 (2021).

Khattab, O. and M. 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”, pp. 39–48 (2020).

Khot, T., P. Clark, M. Guerquin, P. Jansen and A. Sabharwal, “Qasc: A dataset for question answering via sentence composition”, in “Proceedings of the AAAI Conference on Artificial Intelligence”, vol. 34, pp. 8082–8090 (AAAI Press, New York, 2020).

Kim, W., B. Son and I. Kim, “Vilt: Vision-and-language transformer without convolution or region supervision”, in “International Conference on Machine Learning”, pp. 5583–5594 (PMLR, 2021).

Koh, P. W., S. Sagawa, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, T. Lee *et al.*, “Wilds: A benchmark of in-the-wild distribution shifts”, in “International Conference on Machine Learning”, pp. 5637–5664 (PMLR, 2021).

Krishna, R., M. Bernstein and L. Fei-Fei, “Information maximizing visual question generation”, in “IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019”, pp. 2008–2018 (Computer Vision Foundation / IEEE, 2019), URL http://openaccess.thecvf.com/content_CVPR_2019/html/Krishna_Information_Maximizing_Visual_Question_Generation_CVPR_2019_paper.html.

Krishna, R., Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, “Visual genome: Connecting language and vision using crowdsourced dense image annotations”, *International journal of computer vision* **123**, 1, 32–73 (2017).

Kwiatkowski, T., J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le and S. Petrov, “Natural questions: A benchmark for question answering research”, *Transactions of the Association for Computational Linguistics* **7**, 452–466, URL <https://aclanthology.org/Q19-1026> (2019a).

Kwiatkowski, T., J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, M. Kelcey, J. Devlin, K. Lee, K. N. Toutanova, L. Jones, M.-W. Chang, A. Dai, J. Uszkoreit, Q. Le and S. Petrov, “Natural questions: a benchmark for question answering research”, *Transactions of the Association of Computational Linguistics* (2019b).

Kwiatkowski, T., J. Palomaki, O. Redfield, M. Collins, A. P. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey,

M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. V. Le and S. Petrov, “Natural questions: A benchmark for question answering research”, *Transactions of the Association for Computational Linguistics* **7**, 453–466 (2019c).

Laskar, M. T. R., J. X. Huang and E. Hoque, “Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task”, in “LREC”, (Marseille, France, 2020), URL <https://aclanthology.org/2020.lrec-1.676>.

Lee, J., M. Sung, J. Kang and D. 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)”, pp. 6634–6647 (2021).

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

Lee, K., M.-W. Chang and K. Toutanova, “Latent retrieval for weakly supervised open domain question answering”, *ArXiv abs/1906.00300* (2019a).

Lee, K., M.-W. Chang and K. Toutanova, “Latent retrieval for weakly supervised open domain question answering”, in “Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics”, pp. 6086–6096 (Association for Computational Linguistics, Florence, Italy, 2019b), URL <https://aclanthology.org/P19-1612>.

Lewis, P., L. Denoyer and S. Riedel, “Unsupervised question answering by cloze translation”, in “Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics”, pp. 4896–4910 (Association for Computational Linguistics, Florence, Italy, 2019), URL <https://www.aclweb.org/anthology/P19-1484>.

Lewis, P., B. Oğuz, W. Xiong, F. Petroni, W.-t. Yih and S. Riedel, “Boosted dense retriever”, *arXiv preprint arXiv:2112.07771* (2021a).

Lewis, P., E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel and D. Kiela, “Retrieval-augmented generation for knowledge-intensive nlp tasks”, *ArXiv abs/2005.11401* (2020a).

Lewis, P., P. Stenetorp and S. Riedel, “Question and answer test-train overlap in open-domain question answering datasets”, in “EACL”, (2021b).

Lewis, P. S. H., E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel and D. Kiela, “Retrieval-augmented generation for knowledge-intensive NLP tasks”, in “Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual”, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan and H. Lin (2020b), URL <https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html>.

- Li, D., Y. Yang, Y.-Z. Song and T. M. Hospedales, “Learning to generalize: Meta-learning for domain generalization”, in “Thirty-Second AAAI Conference on Artificial Intelligence”, (2018a).
- Li, D., Y. Zhang, H. Peng, L. Chen, C. Brockett, M.-T. Sun and W. B. Dolan, “Contextualized perturbation for textual adversarial attack”, in “Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies”, pp. 5053–5069 (2021).
- Li, G., X. Wang and W. Zhu, “Boosting visual question answering with context-aware knowledge aggregation”, in “Proceedings of the 28th ACM International Conference on Multimedia”, pp. 1227–1235 (2020a).
- Li, L. H., M. Yatskar, D. Yin, C.-J. Hsieh and K.-W. Chang, “Visualbert: A simple and performant baseline for vision and language”, ArXiv preprint **abs/1908.03557**, URL <https://arxiv.org/abs/1908.03557> (2019).
- Li, X., X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei *et al.*, “Oscar: Object-semantics aligned pre-training for vision-language tasks”, in “European Conference on Computer Vision”, pp. 121–137 (Springer, 2020b).
- Li, Y., N. Duan, B. Zhou, X. Chu, W. Ouyang, X. Wang and M. Zhou, “Visual question generation as dual task of visual question answering”, in “IEEE Conference on Computer Vision and Pattern Recognition”, (IEEE Computer Society, 2018b), URL http://openaccess.thecvf.com/content_cvpr_2018/html/Li_Visual_Question_Generation_CVPR_2018_paper.html.
- Li, Y., Y. Li and N. Vasconcelos, “Resound: Towards action recognition without representation bias”, in “Proceedings of the European Conference on Computer Vision (ECCV)”, pp. 513–528 (2018c).
- Li, Y. and N. Vasconcelos, “Repair: Removing representation bias by dataset resampling”, in “Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition”, pp. 9572–9581 (2019).
- Lin, J., X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep and R. Nogueira, “Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations”, ArXiv **abs/2102.10073** (2021).
- Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, “Microsoft coco: Common objects in context”, in “European conference on computer vision”, pp. 740–755 (Springer, 2014).
- Liu, E. Z., B. Haghgoo, A. S. Chen, A. Raghunathan, P. W. Koh, S. Sagawa, P. Liang and C. Finn, “Just train twice: Improving group robustness without training group information”, in “International Conference on Machine Learning”, pp. 6781–6792 (PMLR, 2021).
- Liu, H. and P. Singh, “Conceptnet—a practical commonsense reasoning tool-kit”, BT technology journal **22**, 4, 211–226 (2004).

- Liu, J., L. Cui, H. Liu, D. Huang, Y. Wang and Y. Zhang, “Logiqa: A challenge dataset for machine reading comprehension with logical reasoning”, in “IJCAI”, (2020), URL <https://doi.org/10.24963/ijcai.2020/501>.
- Liu, T.-Y., “Learning to rank for information retrieval”, Foundations and Trends in Information Retrieval **3**, 3, 225–331 (2009).
- Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach”, ArXiv **abs/1907.11692** (2019a).
- Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach”, ArXiv preprint **abs/1907.11692**, URL <https://arxiv.org/abs/1907.11692> (2019b).
- Longpre, S., Y. Lu, Z. Tu and C. DuBois, “An exploration of data augmentation and sampling techniques for domain-agnostic question answering”, in “Proceedings of the 2nd Workshop on Machine Reading for Question Answering”, pp. 220–227 (Association for Computational Linguistics, Hong Kong, China, 2019), URL <https://aclanthology.org/D19-5829>.
- Lopez, L. E., D. K. Cruz, J. C. B. Cruz and C. Cheng, “Transformer-based end-to-end question generation”, ArXiv **abs/2005.01107** (2020).
- Lourie, N., R. Le Bras, C. Bhagavatula and Y. Choi, “Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark”, in “Proceedings of the AAAI Conference on Artificial Intelligence”, vol. 35, pp. 13480–13488 (2021).
- Lu, J., D. Batra, D. Parikh and S. Lee, “Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks”, in “Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada”, edited by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox and R. Garnett, pp. 13–23 (2019), URL <https://proceedings.neurips.cc/paper/2019/hash/c74d97b01eae257e44aa9d5bade97baf-Abstract.html>.
- Luan, Y., J. Eisenstein, K. Toutanova and M. Collins, “Sparse, dense, and attentional representations for text retrieval”, Transactions of the Association for Computational Linguistics **9**, 329–345 (2021).
- Luo, M., “Neural retriever and go beyond: A thesis proposal”, arXiv preprint arXiv:2205.16005 (2022).
- Luo, M., S. Chen and C. Baral, “A simple approach to jointly rank passages and select relevant sentences in the obqa context”, in “Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop”, pp. 181–187 (2022a).
- Luo, M., S. Jain, A. Gupta, A. Einolghozati, B. Oguz, D. Chatterjee, X. Chen, C. Baral and P. Heidari, “A study on the efficiency and generalization of light hybrid retrievers”, arXiv preprint arXiv:2210.01371 (2022b).

- Luo, M., A. Mitra, T. Gokhale and C. Baral, “Improving biomedical information retrieval with neural retrievers”, arXiv preprint arXiv:2201.07745 (2022c).
- Luo, M., Y. Zeng, P. Banerjee and C. Baral, “Weakly-supervised visual-retriever-reader for knowledge-based question answering”, in “Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing”, pp. 6417–6431 (2021).
- Ma, J., I. Korotkov, Y. Yang, K. Hall and R. McDonald, “Zero-shot neural passage retrieval via domain-targeted synthetic question generation”, arXiv preprint arXiv:2004.14503 (2020).
- Ma, J., I. Korotkov, Y. Yang, K. Hall and R. McDonald, “Zero-shot neural passage retrieval via domain-targeted synthetic question generation”, in “Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume”, pp. 1075–1088 (ACL, Online, 2021a).
- Ma, J., I. Korotkov, Y. Yang, K. Hall and R. McDonald, “Zero-shot neural passage retrieval via domain-targeted synthetic question generation”, in “Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume”, pp. 1075–1088 (2021b).
- Ma, J., I. Korotkov, Y. Yang, K. Hall and R. T. McDonald, “Zero-shot neural passage retrieval via domain-targeted synthetic question generation”, in “EACL”, (2021c).
- Ma, X., M. Li, K. Sun, J. Xin and J. 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”, pp. 2854–2859 (2021d).
- Ma, X., K. Sun, R. Pradeep and J. Lin, “A replication study of dense passage retriever”, arXiv preprint arXiv:2104.05740 (2021e).
- MacAvaney, S., A. Yates, A. Cohan and N. Goharian, “CEDR: contextualized embeddings for document ranking”, in “Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019”, edited by B. Piwowarski, M. Chevalier, É. Gaussier, Y. Maarek, J. Nie and F. Scholer, pp. 1101–1104 (ACM, 2019), URL <https://doi.org/10.1145/3331184.3331317>.
- Makino, T. and T. Iwakura, “A boosted supervised semantic indexing for reranking”, in “AIRS”, (Springer International Publishing, Jeju Island, South Korea, 2017).
- Malinowski, M. and M. Fritz, “A multi-world approach to question answering about real-world scenes based on uncertain input”, in “Advances in neural information processing systems”, pp. 1682–1690 (2014).
- Marino, K., X. Chen, D. Parikh, A. Gupta and M. Rohrbach, “Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa”, ArXiv preprint **abs/2012.11014**, URL <https://arxiv.org/abs/2012.11014> (2020).

Marino, K., X. Chen, D. Parikh, A. Gupta and M. Rohrbach, “Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa”, in “Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)”, pp. 14111–14121 (2021).

Marino, K., M. Rastegari, A. Farhadi and R. Mottaghi, “OK-VQA: A visual question answering benchmark requiring external knowledge”, in “IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019”, pp. 3195–3204 (Computer Vision Foundation / IEEE, 2019a), URL http://openaccess.thecvf.com/content_CVPR_2019/html/Marino_OK-VQA_A_Visual_Question_Answering_Benchmark_Requiring_External_Knowledge_CVPR_2019_paper.html.

Marino, K., M. Rastegari, A. Farhadi and R. Mottaghi, “Ok-vqa: A visual question answering benchmark requiring external knowledge”, in “Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition”, pp. 3195–3204 (2019b).

McDonald, R., G. Brokos and I. Androutsopoulos, “Deep relevance ranking using enhanced document-query interactions”, in “EMNLP”, (ACL, Brussels, Belgium, 2018), URL <https://aclanthology.org/D18-1211>.

Mihaylov, T., P. Clark, T. Khot and A. Sabharwal, “Can a suit of armor conduct electricity? a new dataset for open book question answering”, in “Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing”, pp. 2381–2391 (Association for Computational Linguistics, Brussels, Belgium, 2018), URL <https://aclanthology.org/D18-1260>.

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

Miller, G. A., “WordNet: A lexical database for English”, in “Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992”, (1992), URL <https://aclanthology.org/H92-1116>.

Miller, G. A., *WordNet: An electronic lexical database* (MIT press, 1998).

Miller, J. P., R. Taori, A. Raghunathan, S. Sagawa, P. W. Koh, V. Shankar, P. Liang, Y. Carmon and L. Schmidt, “Accuracy on the line: On the strong correlation between out-of-distribution and in-distribution generalization”, in “International Conference on Machine Learning”, pp. 7721–7735 (PMLR, 2021).

Min, S., J. Boyd-Graber, C. Alberti, D. Chen, E. Choi, M. Collins, K. Guu, H. Ha-jishirzi, K. Lee, J. Palomaki, C. Raffel, A. Roberts, T. Kwiatkowski, P. Lewis, Y. Wu, H. Küttler, L. Liu, P. Minervini, P. Stenetorp, S. Riedel, S. Yang, M. Seo, G. Izacard, F. Petroni, L. Hosseini, N. D. Cao, E. Grave, I. Yamada, S. Shimaoka, M. Suzuki, S. Miyawaki, S. Sato, R. Takahashi, J. Suzuki, M. Fajcik, M. Docekal, K. Ondrej, P. Smrz, H. Cheng, Y. Shen, X. Liu, P. He, W. Chen, J. Gao, B. Oguz, X. Chen, V. Karpukhin, S. Peshterliev, D. Okhonko, M. Schlichtkrull,

- S. Gupta, Y. Mehdad and W.-t. Yih, “Neurips 2020 efficientqa competition: Systems, analyses and lessons learned”, in “Proceedings of the NeurIPS 2020 Competition and Demonstration Track”, edited by H. J. Escalante and K. Hofmann, vol. 133 of *Proceedings of Machine Learning Research*, pp. 86–111 (PMLR, 2021), URL <https://proceedings.mlr.press/v133/min21a.html>.
- Min, S., D. Chen, L. Zettlemoyer and H. Hajishirzi, “Knowledge guided text retrieval and reading for open domain question answering”, ArXiv **abs/1911.03868** (2019).
- Mishra, S., A. Arunkumar, C. Bryan and C. Baral, “Our evaluation metric needs an update to encourage generalization”, arXiv preprint arXiv:2007.06898 (2020).
- Mishra, S., D. Khashabi, C. Baral and H. Hajishirzi, “Cross-task generalization via natural language crowdsourcing instructions”, arXiv preprint arXiv:2104.08773 (2021).
- Morris, J. X., E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin and Y. Qi, “Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp”, (2020).
- Mrkšić, N., D. O. Séaghdha, B. Thomson, M. Gašić, L. Rojas-Barahona, P.-H. Su, D. Vandyke, T.-H. Wen and S. Young, “Counter-fitting word vectors to linguistic constraints”, arXiv preprint arXiv:1603.00892 (2016).
- Nam, H., H. Lee, J. Park, W. Yoon and D. Yoo, “Reducing domain gap by reducing style bias”, in “Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition”, pp. 8690–8699 (2021).
- Narasimhan, M., S. Lazebnik and A. G. Schwing, “Out of the box: Reasoning with graph convolution nets for factual visual question answering”, in “Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3–8, 2018, Montréal, Canada”, edited by S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett, pp. 2659–2670 (2018), URL <https://proceedings.neurips.cc/paper/2018/hash/c26820b8a4c1b3c2aa868d6d57e14a79-Abstract.html>.
- Nie, Y., S. Wang and M. Bansal, “Revealing the importance of semantic retrieval for machine reading at scale”, in “Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)”, pp. 2553–2566 (Association for Computational Linguistics, Hong Kong, China, 2019), URL <https://aclanthology.org/D19-1258>.
- Nogueira, R. and K. Cho, “Passage re-ranking with bert”, ArXiv **abs/1901.04085** (2019).
- Nogueira, R., J. Lin and A. Epistemic, “From doc2query to docttttquery”, Online preprint **6** (2019).
- Ozyurt, I. B., A. Bandrowski and J. S. Grethe, “Bio-answerfinder: a system to find answers to questions from biomedical texts”, Database **2020** (2020).

- Papineni, K., S. Roukos, T. Ward and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation”, in “Proceedings of the 40th annual meeting of the Association for Computational Linguistics”, pp. 311–318 (2002).
- Pappas, D., P. Stavropoulos and I. Androutsopoulos, “Aueb-nlp at bioasq 8: Biomedical document and snippet retrieval.”, in “CLEF (Working Notes)”, (2020).
- Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library”, in “Advances in Neural Information Processing Systems 32”, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox and R. Garnett, pp. 8024–8035 (Curran Associates, Inc., 2019), URL <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- Penha, G., A. Câmara and C. Hauff, “Evaluating the robustness of retrieval pipelines with query variation generators”, in “European Conference on Information Retrieval”, pp. 397–412 (Springer, 2022).
- Pruthi, D., B. Dhingra and Z. C. Lipton, “Combating adversarial misspellings with robust word recognition”, in “Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics”, pp. 5582–5591 (2019).
- Qu, C., H. Zamani, L. Yang, W. B. Croft and E. Learned-Miller, “Passage retrieval for outside-knowledge visual question answering”, in “Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval”, pp. 1753–1757 (2021a).
- Qu, Y., Y. Ding, J. Liu, K. Liu, R. Ren, W. X. Zhao, D. Dong, H. Wu and H. Wang, “Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering”, in “Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies”, pp. 5835–5847 (2021b).
- Radford, A., J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision”, arXiv preprint arXiv:2103.00020 (2021a).
- Radford, A., J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision”, in “International Conference on Machine Learning”, pp. 8748–8763 (PMLR, 2021b).
- Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer”, J. Mach. Learn. Res. **21**, 140:1–140:67 (2020).

- Rajpurkar, P., J. Zhang, K. Lopyrev and P. Liang, “SQuAD: 100,000+ questions for machine comprehension of text”, in “Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing”, pp. 2383–2392 (Association for Computational Linguistics, Austin, Texas, 2016), URL <https://www.aclweb.org/anthology/D16-1264>.
- Ramesh, A., M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen and I. Sutskever, “Zero-shot text-to-image generation”, in “International Conference on Machine Learning”, pp. 8821–8831 (PMLR, 2021).
- Rao, J., H. He and J. Lin, “Noise-contrastive estimation for answer selection with deep neural networks”, in “CIKM”, (2016).
- Rao, J., L. Liu, Y. Tay, W. Yang, P. Shi and J. Lin, “Bridging the gap between relevance matching and semantic matching for short text similarity modeling”, in “EMNLP-IJCNLP”, (ACL, Hong Kong, China, 2019), URL <https://aclanthology.org/D19-1540>.
- Reddy, S., D. Chen and C. D. Manning, “CoQA: A conversational question answering challenge”, Transactions of the Association for Computational Linguistics **7**, 249–266, URL <https://aclanthology.org/Q19-1016> (2019).
- Ribeiro, M. T., T. Wu, C. Guestrin and S. Singh, “Beyond accuracy: Behavioral testing of NLP models with CheckList”, in “Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics”, pp. 4902–4912 (Association for Computational Linguistics, Online, 2020), URL <https://www.aclweb.org/anthology/2020.acl-main.442>.
- Robertson, S. and H. Zaragoza, “The probabilistic relevance framework: Bm25 and beyond”, Found. Trends Inf. Retr. **3**, 333–389 (2009a).
- Robertson, S. and H. Zaragoza, “The probabilistic relevance framework: Bm25 and beyond”, Information Retrieval **3**, 4, 333–389 (2009b).
- Robertson, S., H. Zaragoza *et al.*, “The probabilistic relevance framework: Bm25 and beyond”, Foundations and Trends® in Information Retrieval **3**, 4, 333–389 (2009).
- Rohrbach, A., M. Rohrbach, N. Tandon and B. Schiele, “A dataset for movie description”, in “Proceedings of the IEEE conference on computer vision and pattern recognition”, pp. 3202–3212 (2015).
- Rose, S., D. Engel, N. Cramer and W. Cowley, “Automatic keyword extraction from individual documents”, Text mining: applications and theory **1**, 1–20 (2010).
- Saeed, M., N. Ahmadi, P. Nakov and P. Papotti, “RuleBERT: Teaching soft rules to pre-trained lms”, in “EMNLP”, (2021), URL <https://aclanthology.org/2021.emnlp-main.110>.
- Schwartz, R., J. Dodge, N. A. Smith and O. Etzioni, “Green ai”, Communications of the ACM **63**, 12, 54–63 (2020).

- Schwenk, D., A. Khandelwal, C. Clark, K. Marino and R. Mottaghi, “A-okvqa: A benchmark for visual question answering using world knowledge”, arXiv preprint arXiv:2206.01718 (2022).
- Sciavolino, C., Z. Zhong, J. Lee and D. Chen, “Simple entity-centric questions challenge dense retrievers”, in “Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing”, pp. 6138–6148 (2021).
- Seo, M., A. Kembhavi, A. Farhadi and H. Hajishirzi, “Bidirectional attention flow for machine comprehension”, ArXiv **abs/1611.01603** (2017).
- Shah, S., A. Mishra, N. Yadati and P. P. Talukdar, “KVQA: knowledge-aware visual question answering”, in “The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019”, pp. 8876–8884 (AAAI Press, 2019), URL <https://doi.org/10.1609/aaai.v33i01.33018876>.
- Shortliffe, E. H., E. H. Shortliffe, J. J. Cimino and J. J. Cimino, *Biomedical informatics: computer applications in health care and biomedicine* (Springer, 2014).
- Shrivastava, A., A. Gupta and R. Girshick, “Training region-based object detectors with online hard example mining”, in “Proceedings of the IEEE conference on computer vision and pattern recognition”, pp. 761–769 (2016).
- Shrivastava, A. and P. Li, “Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips)”, ArXiv **abs/1405.5869** (2014).
- Singh, A. K., A. Mishra, S. Shekhar and A. Chakraborty, “From strings to things: Knowledge-enabled VQA model that can read and reason”, in “ICCV”, (2019).
- Speer, R., J. Chin and C. Havasi, “Conceptnet 5.5: An open multilingual graph of general knowledge”, in “Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA”, edited by S. P. Singh and S. Markovitch, pp. 4444–4451 (AAAI Press, 2017), URL <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972>.
- Srinivasan, K., K. Raman, J. Chen, M. Bendersky and M. Najork, “Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning”, arXiv preprint arXiv:2103.01913 (2021).
- Sutskever, I., O. Vinyals and Q. V. Le, “Sequence to sequence learning with neural networks”, in “NIPS”, (2014).
- Tan, H. and M. Bansal, “LXMERT: Learning cross-modality encoder representations from transformers”, in “Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)”, pp. 5100–5111 (Association for Computational Linguistics, Hong Kong, China, 2019a), URL <https://aclanthology.org/D19-1514>.

- Tan, H. and M. Bansal, “Lxmert: Learning cross-modality encoder representations from transformers”, arXiv preprint arXiv:1908.07490 (2019b).
- Tan, S., S. Joty, M.-Y. Kan and R. Socher, “It’s morphin’time! combating linguistic discrimination with inflectional perturbations”, in “Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics”, pp. 2920–2935 (2020).
- Taori, R., A. Dave, V. Shankar, N. Carlini, B. Recht and L. Schmidt, “Measuring robustness to natural distribution shifts in image classification”, in “Advances in Neural Information Processing Systems”, vol. 33, pp. 18583–18599 (2020).
- Thakur, N., N. Reimers and J. Lin, “Domain adaptation for memory-efficient dense retrieval”, arXiv preprint arXiv:2205.11498 (2022).
- Thakur, N., N. Reimers, A. Rücklé, A. Srivastava and I. 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)”, (2021a).
- Thakur, N., N. Reimers, A. Rückl'e, A. Srivastava and I. Gurevych, “Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models”, ArXiv **abs/2104.08663** (2021b).
- Tian, J., Y. Li, W. Chen, L. Xiao, H. He and Y. Jin, “Diagnosing the first-order logical reasoning ability through LogicNLI”, in “EMNLP”, (2021), URL <https://aclanthology.org/2021.emnlp-main.303>.
- Tiedemann, J. and S. Thottingal, “OPUS-MT — Building open translation services for the World”, in “Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT)”, (Lisbon, Portugal, 2020).
- Tsatsaronis, G., G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M. Alvers, D. Weissenborn, A. Krithara, S. Petridis, D. Polychronopoulos, Y. Almirantis, J. Pavlopoulos, N. Baskiotis, P. Gallinari, T. Artières, A. N. Ngomo, N. Heino, É. Gaussier, L. Barrio-Alvers, M. Schroeder, I. Androutsopoulos and G. Paliouras, “An overview of the large-scale biomedical semantic indexing and question answering competition”, BMC Bioinformatics **16** (2015).
- Tu, M., K. Huang, G. Wang, J. Huang, X. He and B. Zhou, “Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents”, ArXiv preprint **abs/1911.00484**, URL <https://arxiv.org/abs/1911.00484> (2019).
- Vapnik, V. N. and A. Chervonenkis, “The necessary and sufficient conditions for consistency of the method of empirical risk minimization”, Pattern Recognition and Image Analysis **1**, 3, 284–305 (1991).
- Vaswani, A., N. Shazeer, N. Parmar and et al., “Attention is all you need”, in “NeurIPS”, vol. 30 (Curran Associates, Inc., 2017a), URL <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fdb053c1c4a845aa-Paper.pdf>.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, “Attention is all you need”, Advances in neural information processing systems **30** (2017b).

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, “Attention is all you need”, in “Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA”, edited by I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan and R. Garnett, pp. 5998–6008 (2017c), URL <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fdb053c1c4a845aa-Abstract.html>.

Vrandečić, D. and M. Krötzsch, “Wikidata: a free collaborative knowledgebase”, Communications of the ACM **57**, 10, 78–85 (2014).

Wang, K., N. Thakur, N. Reimers and I. Gurevych, “Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval”, arXiv preprint arXiv:2112.07577 (2021).

Wang, P., Q. Wu, C. Shen, A. Dick and A. Van Den Hengel, “Fvqa: Fact-based visual question answering”, IEEE transactions on pattern analysis and machine intelligence **40**, 10, 2413–2427 (2017).

Wei, J. and K. Zou, “EDA: Easy data augmentation techniques for boosting performance on text classification tasks”, in “Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)”, pp. 6382–6388 (Association for Computational Linguistics, Hong Kong, China, 2019), URL <https://www.aclweb.org/anthology/D19-1670>.

Williams, A., N. Nangia and S. Bowman, “A broad-coverage challenge corpus for sentence understanding through inference”, in “Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)”, pp. 1112–1122 (2018a).

Williams, A., N. Nangia and S. Bowman, “A broad-coverage challenge corpus for sentence understanding through inference”, in “Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)”, pp. 1112–1122 (Association for Computational Linguistics, New Orleans, Louisiana, 2018b), URL <https://aclanthology.org/N18-1101>.

Wu, J., J. Lu, A. Sabharwal and R. Mottaghi, “Multi-modal answer validation for knowledge-based vqa”, ArXiv preprint **abs/2103.12248**, URL <https://arxiv.org/abs/2103.12248> (2021).

Wu, M., N. S. Moosavi, A. Rücklé and I. Gurevych, “Improving qa generalization by concurrent modeling of multiple biases”, in “Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings”, pp. 839–853 (2020).

- Wu, Y., M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, “Google’s neural machine translation system: Bridging the gap between human and machine translation”, arXiv preprint arXiv:1609.08144 (2016).
- Wu, Z. and M. Palmer, “Verbs semantics and lexical selection”, in “Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics”, ACL ’94, p. 133–138 (Association for Computational Linguistics, USA, 1994), URL <https://doi.org/10.3115/981732.981751>.
- Xie, Q., Z. Dai, E. Hovy, T. Luong and Q. Le, “Unsupervised data augmentation for consistency training”, Advances in Neural Information Processing Systems **33** (2020).
- Xiong, C., Z. Dai, J. Callan, Z. Liu and R. Power, “End-to-end neural ad-hoc ranking with kernel pooling”, SIGIR (2017).
- Xiong, L., C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. N. Bennett, J. Ahmed and A. Overwijk, “Approximate nearest neighbor negative contrastive learning for dense text retrieval”, in “International Conference on Learning Representations”, (2020).
- Xu, J., T. Mei, T. Yao and Y. Rui, “Msr-vtt: A large video description dataset for bridging video and language”, in “Proceedings of the IEEE conference on computer vision and pattern recognition”, pp. 5288–5296 (2016).
- Xu, Z., D. Liu, J. Yang, C. Raffel and M. Niethammer, “Robust and generalizable visual representation learning via random convolutions”, in “International Conference on Learning Representations”, (2020).
- Yamada, I., A. Asai and H. 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)”, pp. 979–986 (2021).
- Yang, Y., D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. Hernandez Abrego, S. Yuan, C. Tar, Y.-h. Sung, B. Strope and R. Kurzweil, “Multilingual universal sentence encoder for semantic retrieval”, in “Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations”, pp. 87–94 (Association for Computational Linguistics, Online, 2020), URL <https://aclanthology.org/2020.acl-demos.12>.
- Yang, Y., W.-t. Yih and C. Meek, “WikiQA: A challenge dataset for open-domain question answering”, in “EMNLP”, (ACL, Lisbon, Portugal, 2015), URL <https://aclanthology.org/D15-1237>.
- Yang, Z., Z. Gan, J. Wang, X. Hu, Y. Lu, Z. Liu and L. Wang, “An empirical study of gpt-3 for few-shot knowledge-based vqa”, arXiv preprint arXiv:2109.05014 (2021).

- Yang, Z., P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov and C. D. Manning, “HotpotQA: A dataset for diverse, explainable multi-hop question answering”, in “Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing”, pp. 2369–2380 (Association for Computational Linguistics, Brussels, Belgium, 2018a), URL <https://aclanthology.org/D18-1259>.
- Yang, Z., P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov and C. D. Manning, “Hotpotqa: A dataset for diverse, explainable multi-hop question answering”, in “EMNLP”, (Association for Computational Linguistics, Brussels, Belgium, 2018b).
- Yi, M., L. Hou, J. Sun, L. Shang, X. Jiang, Q. Liu and Z. Ma, “Improved ood generalization via adversarial training and pretraing”, in “International Conference on Machine Learning”, pp. 11987–11997 (PMLR, 2021).
- Yilmaz, Z. A., S. Wang, W. Yang, H. Zhang and J. Lin, “Applying bert to document retrieval with birch”, in “EMNLP/IJCNLP”, (2019).
- Young, P., A. Lai, M. Hodosh and J. Hockenmaier, “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions”, Transactions of the Association for Computational Linguistics **2**, 67–78 (2014).
- Yu, W., Z. Jiang, Y. Dong and J. Feng, “Reclor: A reading comprehension dataset requiring logical reasoning”, in “ICLR”, (2019).
- Yuille, A. L. and C. Liu, “Deep nets: What have they ever done for vision?”, International Journal of Computer Vision **129**, 3, 781–802 (2021).
- Zellers, R., Y. Bisk, A. Farhadi and Y. Choi, “From recognition to cognition: Visual commonsense reasoning”, in “IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019”, pp. 6720–6731 (Computer Vision Foundation / IEEE, 2019), URL http://openaccess.thecvf.com/content_CVPR_2019/html/Zellers_From_Recognition_to_Cognition_Visual_Commonsense_Reasoning_CVPR_2019_paper.html.
- Zellers, R., Y. Bisk, R. Schwartz and Y. Choi, “SWAG: A large-scale adversarial dataset for grounded commonsense inference”, in “Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing”, pp. 93–104 (Association for Computational Linguistics, Brussels, Belgium, 2018), URL <https://www.aclweb.org/anthology/D18-1009>.
- Zhan, J., J. Mao, Y. Liu, J. Guo, M. Zhang and S. Ma, “Jointly optimizing query encoder and product quantization to improve retrieval performance”, in “Proceedings of the 30th ACM International Conference on Information & Knowledge Management”, pp. 2487–2496 (2021).
- Zhang, X., Z. Fang, Y. Wen, Z. Li and Y. Qiao, “Range loss for deep face recognition with long-tailed training data”, in “Proceedings of the IEEE International Conference on Computer Vision”, pp. 5409–5418 (2017).

- Zhang, Z., T. Vu and A. Moschitti, “Joint models for answer verification in question answering systems”, in “Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)”, pp. 3252–3262 (Association for Computational Linguistics, Online, 2021), URL <https://aclanthology.org/2021.acl-long.252>.
- Zhao, T., X. Lu and K. Lee, “Sparta: Efficient open-domain question answering via sparse transformer matching retrieval”, in “Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies”, pp. 565–575 (2021).
- Zhao, Y., X. Ni, Y. Ding and Q. Ke, “Paragraph-level neural question generation with maxout pointer and gated self-attention networks”, in “Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing”, pp. 3901–3910 (Association for Computational Linguistics, Brussels, Belgium, 2018), URL <https://www.aclweb.org/anthology/D18-1424>.
- Zhou, L., C. Xu and J. J. Corso, “Towards automatic learning of procedures from web instructional videos”, in “Thirty-Second AAAI Conference on Artificial Intelligence”, (2018).
- Zhu, Y., O. Groth, M. S. Bernstein and L. Fei-Fei, “Visual7w: Grounded question answering in images”, in “2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016”, pp. 4995–5004 (IEEE Computer Society, 2016), URL <https://doi.org/10.1109/CVPR.2016.540>.
- Zhuang, S. and G. Zuccon, “Characterbert and self-teaching for improving the robustness of dense retrievers on queries with typos”, arXiv preprint arXiv:2204.00716 (2022).

ProQuest Number: 30423007

INFORMATION TO ALL USERS

The quality and completeness of this reproduction is dependent on the quality
and completeness of the copy made available to ProQuest.



Distributed by ProQuest LLC (2023).

Copyright of the Dissertation is held by the Author unless otherwise noted.

This work may be used in accordance with the terms of the Creative Commons license
or other rights statement, as indicated in the copyright statement or in the metadata
associated with this work. Unless otherwise specified in the copyright statement
or the metadata, all rights are reserved by the copyright holder.

This work is protected against unauthorized copying under Title 17,
United States Code and other applicable copyright laws.

Microform Edition where available © ProQuest LLC. No reproduction or digitization
of the Microform Edition is authorized without permission of ProQuest LLC.

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