

UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

Inducing Symbolic Characteristics in Neural Question-Answering Systems via Data Interventions

Permalink

<https://escholarship.org/uc/item/1s16v3b7>

Author

Dua, Dheeru

Publication Date

2023

Copyright Information

This work is made available under the terms of a Creative Commons Attribution-ShareAlike License, available at <https://creativecommons.org/licenses/by-sa/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Inducing Symbolic Characteristics in Neural Question-Answering Systems via Data Interventions

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Dheeru Dua

Dissertation Committee:
Associate Professor Sameer Singh, Chair
Professor Erik Sudderth
Associate Professor Richard Futrell
Senior Research Scientist Matt Gardner

2023

TABLE OF CONTENTS

	Page
LIST OF FIGURES	vi
LIST OF TABLES	x
ACKNOWLEDGMENTS	xiii
VITA	xiv
ABSTRACT OF THE DISSERTATION	xvi
1 Introduction	1
1.1 Problems with Neural QA systems	3
1.1.1 Consistency	3
1.1.2 Decomposability	4
1.1.3 Generalizability	8
1.2 Background	8
1.2.1 Notation	9
1.2.2 Foundation Models	10
1.2.3 Question Answering Models	15
1.2.4 Forms of Reasoning in Question Answering Datasets	18
1.3 Dissertation Outline	20
1.4 Key Findings	21
1.4.1 Imposing Non-IID constraints improves consistency while answering.	21
1.4.2 Generative QA improves consistency while passage retrieval.	22
1.4.3 Purely neural systems lack Compositional reasoning	23
1.4.4 Decomposing complex problem improves few shot model performance	25
1.4.5 Type of dataset shift governs the effectiveness of data augmentation strategies.	27
1.4.6 Sampling data proportional to increase in loss improves generalization	28
1.4.7 Intermediate annotations reduce bias and improve generalization	29
I Improving Consistency: Objectives for Joint Modeling	32
2 Instance Bundles	33
2.1 Introduction	34

2.2	Contrastive Estimation for Reading Comprehension	36
2.2.1	Choosing a neighborhood	38
2.2.2	Alternative uses of bundles	40
2.3	Bundling Heuristics	42
2.4	Experiments	44
2.4.1	Main results	44
2.4.2	Joint Inference	47
2.4.3	Compatibiliy functions for conditional generation models	48
2.4.4	Ablation results	50
2.5	Discussion	50
2.5.1	Relation between MLE, UL and CE	51
2.5.2	The importance of close instance bundles	52
2.6	Related Work	52
3	Generative Question Answering	54
3.1	Introduction	55
3.2	Generative Passage Selection	56
3.2.1	Model Description	57
3.2.2	Model Learning	58
3.3	Experiments and Results	59
3.3.1	Adversarial Evaluation	60
3.3.2	Context pairs vs. Sentences	61
3.4	Conclusion	62
3.5	Related work	63
II	Decomposition: Understanding Compositionality	64
4	DROP: Discrete Reasoning Over Passages	65
4.1	Introduction	66
4.2	Data Collection	68
4.3	Data Analysis	70
4.3.1	Baseline Systems	71
4.3.2	Semantic Parsing	72
4.3.3	SQuAD-style Reading Comprehension	74
4.3.4	Heuristic Baselines	75
4.4	NAQANet	75
4.4.1	Model Description	76
4.4.2	Weakly-Supervised Training	79
4.5	Results and Discussion	79
4.6	Conclusion	81

5	Successive Prompting	86
5.1	Introduction	87
5.2	Decomposing Complex Questions	89
5.2.1	Successive prompting	91
5.2.2	Training paradigm	92
5.3	Synthetic Dataset	93
5.4	Experiments and Results	96
5.4.1	In-context Learning	96
5.4.2	Model Fine-tuning	97
5.4.3	In-context vs Fine-Tuning	100
5.4.4	Qualitative Examples	100
5.5	Related Work	103
5.6	Conclusion	104
III	Generalizability: Challenges and Interventions	105
6	Effectiveness of data augmentation for generalization	106
6.1	Introduction	107
6.2	Background and Evaluation Setup	109
6.2.1	Datasets	110
6.2.2	Models	111
6.3	Generalizability Test	111
6.3.1	Types of dataset shift	112
6.3.2	Calculating shift for ODQA	113
6.3.3	Predicting type of dataset shift	114
6.4	How Well do Models Generalize?	116
6.4.1	Reader Generalization	116
6.4.2	Retriever Generalization	117
6.5	Interventions for Improving Adaptation	118
6.5.1	Zero-shot adaptation methods	119
6.5.2	Few-shot Generalizability and Adapatability	122
6.6	Related Work	124
6.7	Appendix	125
6.7.1	How are evaluation sets curated?	125
6.7.2	Varying context distribution	126
6.7.3	Varying answer distribution and pre-training corpus	127
6.7.4	Degree of domain shift	128
6.7.5	Statistical Significance	128
6.7.6	Qualitative examples generated by the few-shot method	128
7	ORB: Open-Reading Benchmark	130
7.1	Introduction	131
7.2	Datasets	132
7.3	Synthetic Augmentations	135

7.4	Experiments	138
7.4.1	Model	138
7.4.2	Existing Dataset Performance	139
7.4.3	Synthetic Augmentations	140
7.5	Related Work	140
7.6	Conclusion	142
8	Dynamic Sampling for improving multi-task generalization	144
8.1	Introduction	145
8.2	Sampling and Scheduling Strategies	146
8.2.1	Instance Sampling	147
8.2.2	Epoch Scheduling	148
8.3	Experiments	150
8.4	Conclusions	152
9	Benefits of Intermediate Annotations	154
9.1	Introduction	155
9.2	Intermediate Annotations	156
9.3	Experiments and Results	158
9.3.1	Setup	158
9.3.2	Cost vs Benefit	160
9.3.3	Bias Evaluation	161
9.3.4	Qualitative Result	163
9.3.5	Related Works	164
9.4	Conclusion	165
10	Conclusion and Future Directions	166
	Bibliography	169

LIST OF FIGURES

	Page
1.1 Example from BioASQ: Grounding objects in context for answering question . . .	2
1.2 Inconsistency: Example from HotpotQA dataset where neural systems produce same answer for opposite questions.	4
1.3 Answer retrieved by a traditional search engine. These results are from Google ¹ as observed on May 8, 2023	5
1.4 A large language model is unable to filter items for hyperthyroidism. These results are from ChatGPT ² on May 8, 2023	6
1.5 Retrieval augmented language models are unable to perform the right reasoning. These results are from Bing’s chat feature on May 8, 2023	6
1.6 Retrieve-and-Read models trained on general-purpose domain with Wikipedia corpus and tested with Wikipedia and Pubmed corpus.	9
1.7 Transformer Block [1]	11
1.8 Encoder-Decoder Transformer Architecture	13
1.9 Question Answering as span selection problem: Probability distribution of start and end of answer spans, as heat map over words in the passage.	16
1.10 Question Answering as generation problem: At each step (i), the next token is generated based on representations from input tokens and previously generated tokens	17
1.11 The instructions explain the task to be performed in natural language, and then examples (in blue) demonstrate how to solve examples analogous to the test example (in pink). The model is prompted with “Answer:” to continue producing the answer to the given question.	18
1.12 Reasoning types in QA datasets: The first two are simple questions while the last question is compositional	19
1.13 Performance (F1) on HotpotQA dev set with IID vs non-IID inference for a model trained with MLE vs CE.	22
1.14 Performance (Acc) comparison of discriminative vs generative model with original and adversarial dev set.	23
1.15 Successive prompting exploits shared sub-problems from questions that are not similar on the surface	24
1.16 Purely Symbolic and Neural systems do not perform as well as Neuro-Symbolic (Hybrid) systems.	25

¹www.google.com

²<https://chat.openai.com/>

1.17	Successive prompting exploits shared sub-problems from questions that are not similar on surface	26
1.18	Decomposing complex problems into simpler problems allows better generalization.	27
1.19	Effect of data augmentation techniques on dataset shifts	28
1.20	Baseline vs. Dynamic sampling averaged over low, medium and high resource datasets in ORB	29
1.21	For the same cost, intermediate annotations help diffuse biased over-representation of number 3 as compared to adding more question-answer pairs	30
2.1	Instance bundle created from HotpotQA	34
2.2	Probability of gold QA pair normalized over all questions in the bundle. The higher value indicates that positive QA pair has a high likelihood and at the same time negative QA pair has a low likelihood. At 0.5, both the contrastive questions would produce the same answer with the same likelihood. In an ideal scenario, the distribution should be a delta function at 1.0	34
2.3	Contrastive Estimation models. In each subfigure an instance bundle of size 2 is shown, with bold lines indicating combinations whose probability is maximized at the expense of the combinations represented by gray lines, for the positive QA pair (q_0, a_0) in the bundle. The total CE loss is the sum of loss for each positive QA pair in the bundle.	38
2.4	Performance (F1) on COMPARISON dev with independent prediction versus joint inference. Joint inference improves all models. The results are on the subset of COMPARISON for which we have paired instances (~93%).	48
2.5	Answer decoder	49
3.1	Example from HotpotQA, showing the reasoning chain for answering the question (in green) and an adversarial context (in pink) introduced by [2] which confuses the model, causing it to change its prediction because it did not learn the right way to reason.	56
4.1	Distribution of the most popular question prefixes for two different subsets of the training data.	71
4.2	Numerically-augmented QANet. The likelihood of answer being “3” is $0.42p + 0.1(s_1 + s_2 + s_3) + 0.10c + 0.38q$	78
4.3	Question Answering HIT sample above with passage on the left and input fields for answer on the right and Highlighted candidate spans of sample answers below . . .	85
5.1	Example decomposition used by Successive Prompting’s question decomposition and question answering stage on a DROP example. The model iterates between predicting a simple question to ask and answering the simple question.	87
5.2	A demonstration of successive prompting with in-context learning. The selected examples for supervision and complex question to be answered pre-pended with the context paragraph (omitted to simplify illustration) are encoded by the model to generate question and answer at QD and QA stage respectively. During fine-tuning, only training supervision is used in an i.i.d manner for learning QD and QA models.	90

5.3	Examples of COUNT operation and some of its higher order combinations, with natural language and symbolic decompositions of the complex question. Underneath the first instance of a symbolic operation we show its corresponding natural language version. See Table 5.1 for the original table used to generate context and questions.	95
5.4	Generated decompositions depicting strength and weaknesses of Successive Prompting.	101
6.1	Effect of interventions on dataset shifts. <i>Top:</i> Average end-to-end performance of source domain model is quite poor when applied to OOD datasets. Source model (trained on general-purpose domain) performance improves when adapted to unseen target domain with interventions. <i>Bottom:</i> Drill-down of performance into zero and few-shot data augmentations averaged over target datasets exhibiting these shifts shows covariate and concept shifts respond to zero and few-shot data augmentations. Target datasets with No shift do not improve much with any intervention while full shift benefits most from Few-shot.	108
6.2	Generalizability Test: At the first level, we decide whether the input distribution is closer to the uniform distribution or gold. At the second level, the gradual increase from left to right in the leaf nodes depicts decrease in distance of output distribution from uniform. The target datasets at the bottom are placed based on distances in Table 6.1. The nodes represent if the source model $p_s(a q, c)$ is compatible or not with the target dataset $p_t(a q, c)$.	112
6.3	Reader performance on the target set without any interventions. SearchQA, Quasar-S and Quasar-T do not have gold passage annotations and so UB-READ does not improve over UB-RET. The majority voting baseline on COLIEE is 50.95.	118
6.4	Retriever performance (Acc@100) without any interventions on target domain corpus	119
6.5	Retriever Performance (Acc@100): Varying context distribution by creating a combined document index. For COLIEE, we use oracle passages for performance computation.	126
6.6	Reader Performance (F1): Effect of change in context distribution with BM25 retrievals from the combined index.	127
6.7	Examples of data generated from few-shot prompting.	129
8.1	Illustration of Epoch Scheduling Strategies with Dynamic Sampling. Instances are sampled dynamically in proportion to exact match accuracy difference of 25%, 10% and 15% for task 1, 2 and 3 respectively. M1, M2, ... M9 depict nine mini-batches in an epoch.	149
9.1	Example from DROP, showing the intermediate annotations that we collected via crowd-sourcing.	155
9.2	Example collected annotation from Quoref, showing the intermediate steps.	157
9.3	Performance of model for a varying percentage of budget invested in collecting intermediate annotation. The calculation was done with costs as \$0.4 and \$0.7 for a QA pair in DROP and Quoref, respectively.	159
9.4	Performance of model trained on varying amount of annotations used in training	160

9.5	For the same cost intermediate annotations helps diffuse biased over-representation of number 3 as compared to adding more question-answer pairs	162
9.6	For the same cost intermediate annotations helps diffuse biased over-representation of number 3 as compared to adding more question-answer pairs	163
9.7	Predicted relevant spans for question answered correctly with annotation (prediction: “3”) and incorrectly without annotations (prediction: “2”) by MTMSN model trained on DROP	163
9.8	Predicted relevant spans for question answered correctly with annotation (prediction: “Charles Spencer Cowper”) and incorrectly without annotations (prediction: “Lord Palmerston”) by XLNet on Quoref	164
9.9	HIT interface used for collection annotations	165

LIST OF TABLES

	Page
1.1 Example questions and answers from the drop dataset, showing the relevant parts of the associated passage and the reasoning required to answer the question.	7
2.1 COMPARISON dev set performance	45
2.2 F1 performance on full HotpotQA dev set with increasing number (k) of top-k negative answer candidates	46
2.3 ROPES dev set performance	46
2.4 Performance (EM/F1) of COMPARISON dev set on models finetuned with different single and multiple neighborhood CE losses.	50
2.5 Performance (EM) of ROPES dev set on models finetuned with different single and multiple neighborhood CE losses. We do not report F1 as it’s not a good performance measure on this dataset	50
2.6 Comparison between QUOREF, COMPARISON, and ROPES datasets with Top-k bundling. MLE, UL, and CE results are on the corresponding development sets (F_1 for COMPARISON and QUOREF, EM for ROPES) and Entropy ₁₀ and Top-2 ratio are measured on random samples of the training sets. UL and CE columns show results after fine-tuning the baseline MLE model with the respective objectives.	53
3.1 Passage selection accuracy: Accuracy that the selected passage pair (c_{ij}^*) by different techniques is the oracle one (c_{gold}) on original development set.	59
3.2 Performance on Adversarial Data: Passage selection accuracy and end to end QA F1 on original and adversarial set [2] of HotpotQA. The results of [3] and [4] are taken from [5].	60
3.3 Sample questions generated by using the question generation decoder with top-k sampling show that the generative model is able to construct (reason about) possible multi-hop questions given a context-pair.	61
3.4 Passages vs Sentences: Passage selection accuracy for models with different context inputs on the development and adversarial set of HotpotQA.	62
4.1 Example questions and answers from the drop dataset, showing the relevant parts of the associated passage and the reasoning required to answer the question.	82
4.2 Dataset statistics across the different splits.	83
4.3 Distribution of answer types in training set.	83

4.4	Performance of the different models on our development and test set, in terms of Exact Match (EM), and numerically-focused F_1 (§4.3.1). Both metrics are calculated as the maximum against a set of gold answers.	83
4.5	Representative examples from our model’s error analysis. We list the identified semantic phenomenon, the relevant passage highlights, a gold question-answer pair, and the erroneous prediction by our model.	84
4.6	Dev set performance breakdown by different answer types; our model (NAQANet, marked as $QN+$) vs. BERT, the best-performing baseline.	84
5.1	Example table from Wikipedia where rows become sentences and columns are used for question generation (used as context for Figure 5.3).	96
5.2	F1 Performance of in-context prompting on the DROP dev set with and without in-domain annotations.	97
5.3	F1 Performance of various model architectures on DROP dev-set pre-trained on synthetic data and further fine-tuned with 300 DROP examples.	99
5.4	F1 with QD and QA modules from incontext learning and fine-tuning with only DROP annotations	100
6.1	Wasserstein distance computed over 100 labeled examples from the target set. The negative retriever value implies that the target dataset falls on the right side of decision tree at first level (Figure 6.2).	114
6.2	Answer distribution: Retriever (DPR) and Reader (FiD with BM25 retrievals) F1 on BioASQ.	120
6.3	Zero-shot: Comparing retriever (DPR) and reader (FiD with BM25 retrievals) performance on two types of question formats for augmentation.	121
6.4	Both Closed Book and DataGen use eight few-shot examples from the target domain. Closed Book LLM contains 540B params while the Retriever and Reader contain 110M and 770M params respectively. Closed-book performance for NQ is 36.71.	122
6.5	Answer distribution: Reader performance on BioASQ with C4 and Pubmed pre-trained T5	127
6.6	Wasserstein distance computed over 100 target domain examples. The distance between reference (source) and uniform over 100 validation set source domain examples is $v_u^r=0.2925$	128
7.1	Dataset Statistics	134
7.2	Yields of augmented datasets	136
7.3	Examples of generated augmentations with various templates.	137
7.4	Quality of augmented datasets (# of good questions out of 50 sampled)	138
7.5	Performance on baseline BERT model on different datasets	139
7.6	Quantitative and qualitative analysis of generated augmentations. We only show performance for high yield and high-quality augmentations.	141
8.1	Open Reading Benchmark (ORB) Datasets	150
8.2	Effect of using different instance sampling strategies with heterogeneous batch scheduling	150
8.3	Effect of using different epoch scheduling strategies with dynamic sampling	151

8.4 Results on ORB test sets. 151

9.1 F1 performance and confusion loss (lower is better) of models in three settings: baseline with 10k(DROP) and 5k(Quoref) QA pairs, additional QA pairs worth \$250 and \$100 for DROP and Quoref respectively, and additional annotations worth \$250 and \$100 for DROP and Quoref respectively. To put confusion loss in perspective, the *best* confusion loss, i.e. perfect diffusion, is 90.1 for DROP and 87.0 for Quoref. 162

ACKNOWLEDGMENTS

I am extremely grateful to my advisor, Sameer Singh, for providing me this great opportunity to learn and develop as an independent researcher. He always gave me freedom to explore diverse directions and taught me to embrace the concept of failing fast. I would also like to express my gratitude towards Matt Gardner, who sparked my passion for research and has been part of my journey in the United States from the beginning. He helped me develop critical thinking and recognize the gaps in my knowledge. I have been extremely privileged to have been mentored by these two exceptional individuals who not only taught me what it means to do research but also instilled in me the importance of prioritizing happiness in life. I would also like to thank Erik Sudderth and Richard Futrell for their time and being part of my defense committee.

I have also been extremely fortunate to have worked with so many amazing people during internships at Amazon, Meta AI, and Google Research. First, I would like to thank Cicero Nogueira dos Santos for being a great mentor during my time at IBM and Amazon and always being a call away whenever I needed advice. Second, I would like to extend my gratitude to Angela Fan and Antoine Bordes, whose support and encouragement during a pivotal phase in my life greatly shaped my life for the better. I would also like to thank Mike Lewis for imparting invaluable knowledge and teaching me how to debug neural systems, which significantly improved my understanding. Finally, I would like to thank Pat Verga, Emma Strubell, and William Cohen for giving me an incredible opportunity to be a part of an exciting and collaborative work environment. William is a kind and humble teacher who always prioritizes the well-being and success of his students. Pat is an incredible mentor who not only appreciates challenges but also consistently motivates and propels me to strive for continuous improvement.

I would also like to thank my co-authors and UCI-NLP lab mates for wonderful discussions: Pradeep Dasigi, Gabriel Stanovsky, Yizhong Wang, Shruti Bhosale, James Cross, Alon Talmor, Ananth Gothumukkala, Shivanshu Gupta, Bing Xiang, Kevin Gimpel, Niranjana Balasubramanian, Patrick Ng, Zhengli Zhao, Nitish Gupta, Ana Marasovic, Robert Logan, Qiang Ning, Vedanuj Goswami, Ben Athiwaratkun, Jun Seok Kang, Ananya, Yasaman Razhegi, Catarina Belem, Tamanna Hossain-Kay, Preethi Seshadri, Anthony Chen, Kolby Nottingham and Dylan Slack.

On a personal note, I would like to thank my parents for always being supportive of my decisions and encouraging me to pursue my dreams. Finally, I would like to thank my husband, Rohan, for his unwavering support, love, care and Yoda-esque talks, especially in moments of self-doubt and setbacks.

The material presented in this dissertation was funded in part by: Department of the Air Force/Air Force Misc Bases (446638-22504), National Science Foundation CRII (IIS-1756023), DARPA MCS program under Contract No. N660011924033 with the United States Office Of Naval Research, Amazon via Alexa (00181440-2019-TR), Fellowship by Hasso Plattner Institute (HPI), the Allen Institute of Artificial Intelligence, and Google. The views expressed are those of the author and do not reflect the official policy or position of the funding agencies.

VITA

Dheeru Dua

EDUCATION

Doctor of Philosophy in Computer Science University of California	2023 <i>Irvine, CA</i>
Masters in Intelligent Information Systems Carnegie Mellon University	2015 <i>Pittsburgh, PA</i>
Bachelor of Technology in Computer Science Indira Gandhi Institute of Technology	2011 <i>Delhi, India</i>

RESEARCH EXPERIENCE

Graduate Research Assistant University of California, Irvine	2017–2023 <i>Irvine, California</i>
--	---

TEACHING EXPERIENCE

Teaching Assistant University of California	2017–2018 <i>Irvine, CA</i>
---	---------------------------------------

PUBLICATIONS

To Adapt or to Annotate: Challenges and Interventions for Domain Adaptation in Open-Domain Question Answering [6] Association for Computational Linguistics	Jul 2023
Successive Prompting for Decomposing Complex Questions [7] Empirical Methods for Natural Language Processing	Dec 2022
Tricks for Training Sparse Translation Models [8] North American Association for Computational Linguistics	Jun 2022
Learning with instance bundles for reading comprehension [9] Empirical Methods for Natural Language Processing	Sep 2021
Generative Context Pair Selection for Multi-hop Question Answering [10] Empirical Methods for Natural Language Processing	Sep 2021

Evaluating models' local decision boundaries via contrast sets [11] Empirical Methods for Natural Language Processing	Jul 2020
Easy, reproducible and quality-controlled data collection with CROWDAQ [12] Empirical Methods for Natural Language Processing	Jun 2020
Benefits of intermediate annotations in reading comprehension [13] Association for Computational Linguistics	Jul 2020
Dynamic sampling strategies for multi-task reading comprehension [14] Association for Computational Linguistics	Jul 2020
ORB: An open reading benchmark for comprehensive evaluation of machine reading comprehension [15] Machine Reading for Question Answering Workshop	Jul 2020
PoMo: Generating entity-specific post-modifiers in context [16] North American Association for Computational Linguistics	Jun 2019
DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs [17] North American Association for Computational Linguistics	Jun 2019
Generating natural adversarial examples [18] International Conference on Learning Representations	May 2018

INDUSTRY EXPERIENCE

Research Intern Google Deepmind	2022 <i>Pittsburgh, PA</i>
Research Intern Meta AI Research	2021 <i>Palo Alto, CA</i>
Research Intern AWS Research, Amazon	2020 <i>New Yorck City, NY</i>
Cognitive Research Engineer IBM Research	2015–2017 <i>Yorktown Heights, NY</i>
Software Engineer Microsoft R&D Center	2011–2014 <i>Hyderabad, India</i>

ABSTRACT OF THE DISSERTATION

Inducing Symbolic Characteristics in Neural Question-Answering Systems via Data Interventions

By

Dheeru Dua

Doctor of Philosophy in Computer Science

University of California, Irvine, 2023

Associate Professor Sameer Singh, Chair

Question Answering (QA) is an innate way in which humans converse, understand abstract concepts, and reason about complex interactions among different objects. This makes question answering a great way to test the natural language understanding of an artificial intelligence system. The recent advances in model architectures and large-scale datasets have led to the development of neural QA systems that surpass human performance on simple question answering – which requires looking up a single fact in given a document. The reason behind the success of neural systems lies in their ability to directly learn features to extract answers from data. In contrast, symbolic systems encounter notable difficulties in scaling due to their restricted applicability to semi-structured or symbol-grounded data. Despite their reliance on structured data, symbolic systems demonstrate proficiency in executing deterministic operations and performing reasoning tasks. Conversely, neural systems exhibit limitations in reasoning, as they are (1) inconsistent, (2) unable to compose simple facts and perform complex reasoning, and (3) sensitive to changes in domain distribution.

In this dissertation, we present a range of data intervention schemes that facilitate in building consistent, decomposable, and generalizable neural QA systems. In the first part, we show that purely neural systems are inconsistent and biased because most training and data collection procedures for neural systems make the independence assumptions. We explore two ways to address this problem in the context of question answering. Firstly, we introduce a way to curate related QA pairs and

a contrastive loss function that takes bundles of related questions as input and learns to jointly answer them. Such related or minimally different data enable us to capture the relationship across examples that would otherwise have remained unseen or, at the very least, not seen simultaneously at the time of training. This method improves the performance over baseline methods by $\sim 14\%$ in F1. Secondly, we take a generative approach to passage selection in multi-hop QA, where we learn a prior over interactions between the current passage and the subsequent passage for hopping along with a conditional question generation model, which ensure that parts of the question are not overlooked. This improves the performance over baseline on adversarial set by $\sim 5\%$ in passage selection accuracy.

In the second part, we first introduce a compositional QA dataset, DROP, that requires extracting multiple facts from a given document and then performing discrete (symbolic) operations on these facts while solving a question. We show that purely neural and symbolic methods perform poorly on this task – even commercial systems like GPT-4 are unable to reach human performance. Then, we show that prompting-based neural QA methods lack decomposability for two reasons: (1) the intermediate steps are not specified in a common format across different complex questions, (2) the model ignores valuable sub-problem demonstrations associated with complex questions that do not appear similar on the surface to the question at hand, even though they provide much better supervision signal. We propose a method that leverages synthetically generated data to break down a complex question into simpler sub-question answer pairs in a consistent format. This data intervention not only promotes transfer but also enables us to search for pertinent QA pairs at each step especially tailored to solving that step, further improving supervision signal and overall performance in a zero-shot setting by $\sim 5\%$.

Finally, we dissect the complex interactions among questions, answers, and documents learned by a neural QA system to assess their effectiveness towards generalization under a range of different data distributions. We demonstrate three distinct data intervention techniques that aid in improving generalization capabilities. The first data intervention approach involves zero-shot and few-shot

adaptation methods that use none to little examples from unseen domain to effectively adapt an existing QA model to those unseen domains. Our few-shot method improves out-of-domain generalization performance by up to $\sim 24\%$. This style of intervention is typically employed for adapting to a specific target domain. However, in real-world scenarios, a single model needs to possess the capability to learn from multiple domains. To this end, we introduced an open reading evaluation benchmark (ORB) that tests a single model performance on a range of reasoning abilities. We show that the models have a tendency to forget how to solve examples that they were previously capable of solving, when trained on multiple datasets. Our second intervention technique revolves around strategic sampling of examples from various datasets to alleviate catastrophic forgetting. This method, called dynamic sampling improves multi-task performance by $\sim 13\%$. In the final data intervention technique, we introduce intermediate reasoning steps when solving compositional problems. This approach reduces label bias by up to $\sim 61\%$. Overall, we demonstrate how data interventions can be utilized to induce characteristics of symbolic systems into neural QA systems.

Chapter 1

Introduction

Thinking is manipulation of symbols and
Reasoning is computation

Thomas Hobbes

(grandfather of Artificial Intelligence)

Question Answering (QA) remains one of the primary tasks for testing human [19] and machine reading [20] capabilities. This is because the reader has to not only understand the linguistic constructs of the natural language of the text but also reason about multiple facts that require an intimate understanding of the ways of the world.

Early attempts at building question answering systems [21, 22] use symbolic logic over manually written rules and/or facts that describe the world. The research community, over the years, has developed large-scale knowledge graphs that specify the world through symbols. These symbols capture the abstract meaning of various concepts like events [23, 24], entities [25, 26, 27] and commonsense reasoning [28] through description about these concept and/or how a specific concept relates to other concepts. In cognitive science, this problem of how a word or symbol gets its meanings is referred to as symbol grounding [29, 30]. Symbol grounding can be challenging in

natural language because writers do not often mention the meaning of a concept and assume that readers have an inherent understanding of how to ground a symbol in the real world and envision the effects of manipulations performed on these symbols. For instance, to answer the question in Figure 1.1, the reader needs to: (1) have domain knowledge that **SIRT1** is a protein (2) be able to perform symbol manipulation and comprehend that **EMT** refers to **epithelial-mesenchymal transition** in the given context and (3) be able to reason that **silencing protein SIRT1 restores cell-cell adhesion** implies that presence of SIRT1 causes the loss of cell-cell adhesion further resulting in transition of epithelial cells into mesenchymal cells.

Context: We propose a novel mechanism through which SIRT1 regulates **EMT** in prostate cancer cells through cooperation with the EMT inducing transcription factor ZEB1. We found that forced expression of SIRT1 in non-transformed prostate epithelial cells disrupts the epithelial morphology concomitant with decreased expression of the epithelial marker, E-cadherin, and increased expression of mesenchymal markers. In contrast, **silencing SIRT1 in metastatic prostate tumor cells restores cell-cell adhesion** and induces a shift toward an epithelial morphology concomitant with increased expression of E-cadherin and decreased expression of mesenchymal markers.

Question: Which protein causes the loss of cell-cell adhesion during **epithelial-mesenchymal transition**?

Answer: **SIRT1**

Figure 1.1: Example from BioASQ: Grounding objects in context for answering question

While knowledge graphs facilitate symbol grounding and manipulation, they are expensive to curate making them limited and difficult to extend beyond the domains for which such facts are specified. Additionally, the ambiguous, nuanced, and compositional nature of natural language poses a lot of challenges in parsing natural language into a form over which symbolic logic can be executed [31].

Neural networks, conversely, learn to extract facts and/or rules by themselves from examples demonstrating how to solve a task, making them extensible. In fact, with recent advances in research, neural networks can learn associations between various concepts without human intervention – by analyzing concept co-occurrences in existing cheap and abundant unstructured natural language data [32, 33]. This paradigm is called pre-training, where we learn latent representations for concepts that appear in natural language text. Several works [34, 35] even use multi-sensory data

from images, text, speech, etc., to jointly learn latent representations of various symbols; and have been shown to have a better understanding of a concept.

Besides the representation of symbols or concepts, QA systems face the additional challenge of performing reasoning to solve the task, particularly when dealing with intricate compositional problems [36]. Symbolic systems conduct deterministic reasoning, which results in consistent outputs. Notably, they can also be applied to unseen symbolic compositions in a generalizable manner. In contrast, neural networks are stochastic, and their learning is highly influenced by the distribution of the data, which can lead to unanticipated biases [13, 2] and limited generalization [37, 38, 39].

Given that neural systems are more adept at learning symbol representation and symbolic systems excel in reasoning, we take inspiration from symbolic systems and focus on alleviating three problems of neural question answering systems: consistency, decomposability, and generalizability with the help of various data interventions.

1.1 Problems with Neural QA systems

We will look at three problems with neural systems in detail below.

1.1.1 Consistency

Recent advancements in QA systems have led to a human-level performance on simple questions that require extracting answers in a given passage¹². However, it is unclear if they can consistently solve such questions when deployed for human use. In Fig.1.2, we show that a model trained on HotpotQA [40] is able to answer question q_1 correctly, but when we replace the word “more” with

¹<https://rajpurkar.github.io/SQuAD-explorer/>

²<https://hotpotqa.github.io/>

“less”, it gets confused. Inconsistent responses lead to a lack of trust and credibility with users. It is imperative for any user system not to get confused with simple and straightforward changes to the input [41]. In an ideal scenario, if a model knows the answer to q_1 , it should automatically know the answer to q_2 and vice-versa.

In symbolic systems, we can specify the causal relation between such words with a simple logic rule like more $(m_1, m_2) \implies$ less (m_2, m_1) . Neural systems, however, need to learn that the words more and less are not only related but are semantically polar opposites of each other from examples alone. And they need to acquire such associations from examples that are not even presented together during training. Neural systems are generally trained with the assumption that data is independently and identically distributed (IID), which breaks correlations between examples that are closely related leading to inconsistent learning [9].

Context: Marsilea is a genus of approximately 65 species of aquatic ferns of the family Marsileaceae. The name honours Italian naturalist Luigi Ferdinando Marsili (1656-1730) Brabejum is a genus of a single species of large evergreen tree, Brabejum stellatifolium in the Proteaceae, commonly called wild almond, bitter almond, or ghoeboontjie.

Question (q_1): Is the Marsilea or the Brabejum, the genus of more individual species of plants?
Answer (a_1): Marsilea

Question (q_2): Is the Marsilea or the Brabejum, the genus of less individual species of plants?
Answer (a_2): Marsilea

Figure 1.2: Inconsistency: Example from HotpotQA dataset where neural systems produce same answer for opposite questions.

1.1.2 Decomposability

Humans frequently solve problems by breaking down intricate tasks, recognizing recurring patterns, and gradually moving toward the solution through incremental steps. Figure 1.3 shows a response to an everyday question a person might ask a search engine like Google. A search engine, being a simple retrieval-based system, is able to identify suitable food options for people with overactive thyroid or hyperthyroidism. However, it does not possess the capability to reason and filter the list

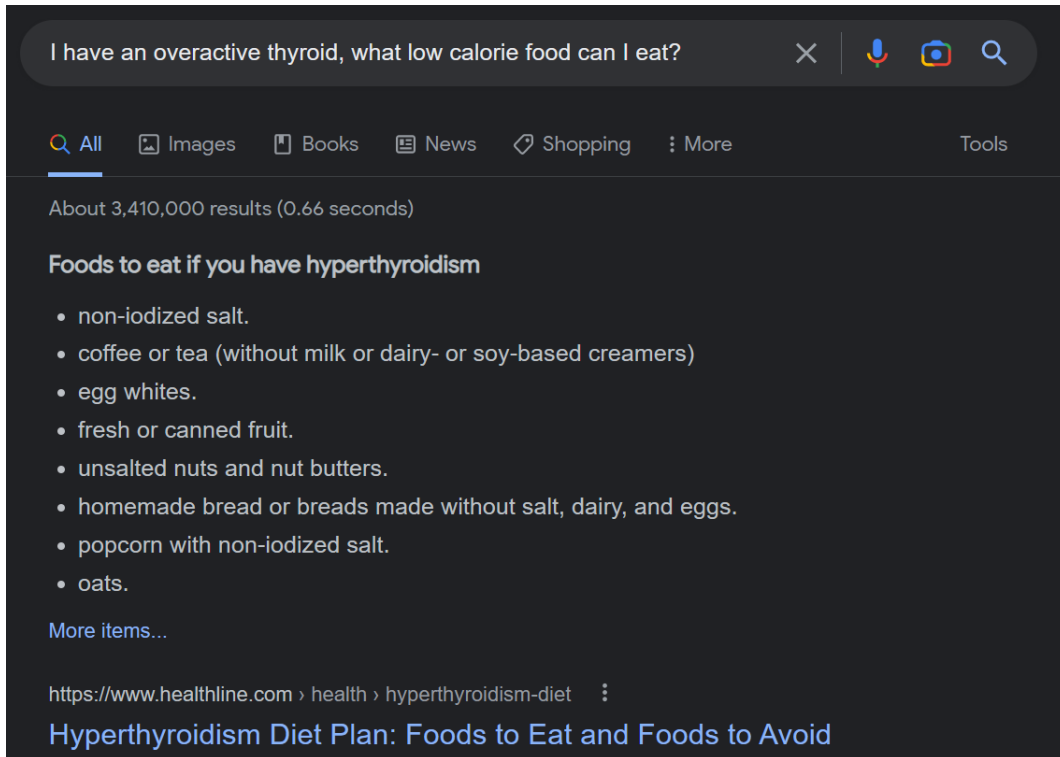


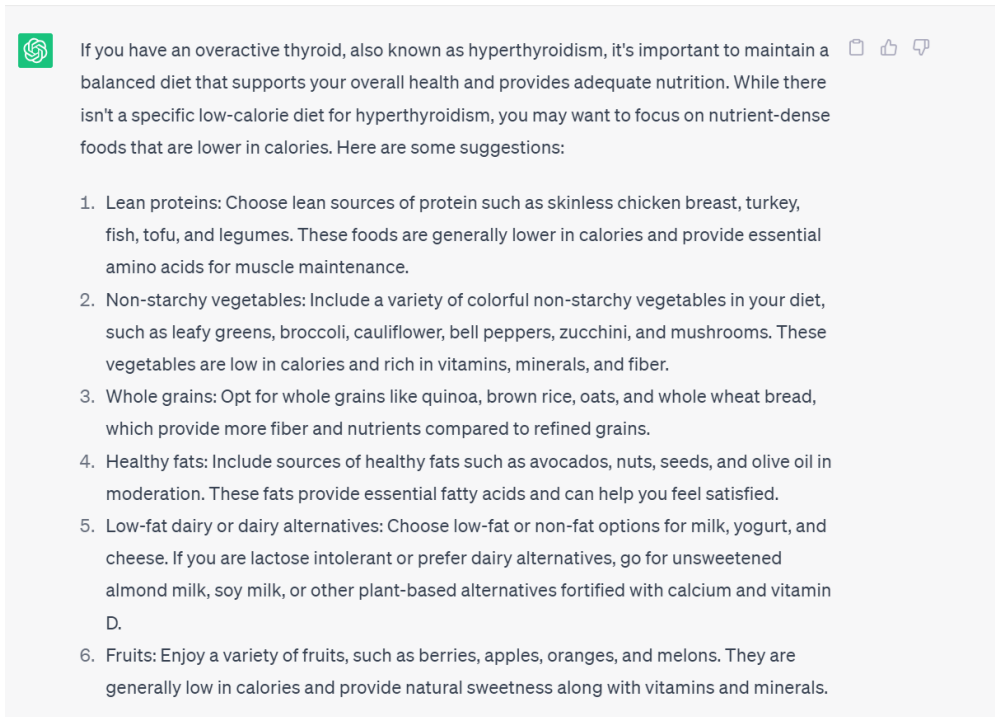
Figure 1.3: Answer retrieved by a traditional search engine. These results are from Google³ as observed on May 8, 2023

of food items based on their calorie content and concludes that "unsalted nuts and nut butters" have a low calorie count.

On the other hand, large language models [42] (LLM), which are considered to be proficient at performing general-purpose reasoning out-of-the-box, are able to identify low-calorie foods, as shown in Figure 1.4. But they are unable to perform accurately narrow down the list to only include the items suitable for individuals with hyperthyroidism. This is evident from their recommendation of soy products like tofu that are not advisable for individuals with an overactive thyroid and also healthy fats that are not low in calories. In fact, even a combination of a search engine and a large language model is unable to perform the right reasoning and filter out high calorie foods "Unsalted nuts and nut butters" as shown in Figure 1.5.

Symbolic systems can effectively handle compositions via semantic parsing style approaches or inference techniques like forward and backward chaining, provided the facts and rules are specified

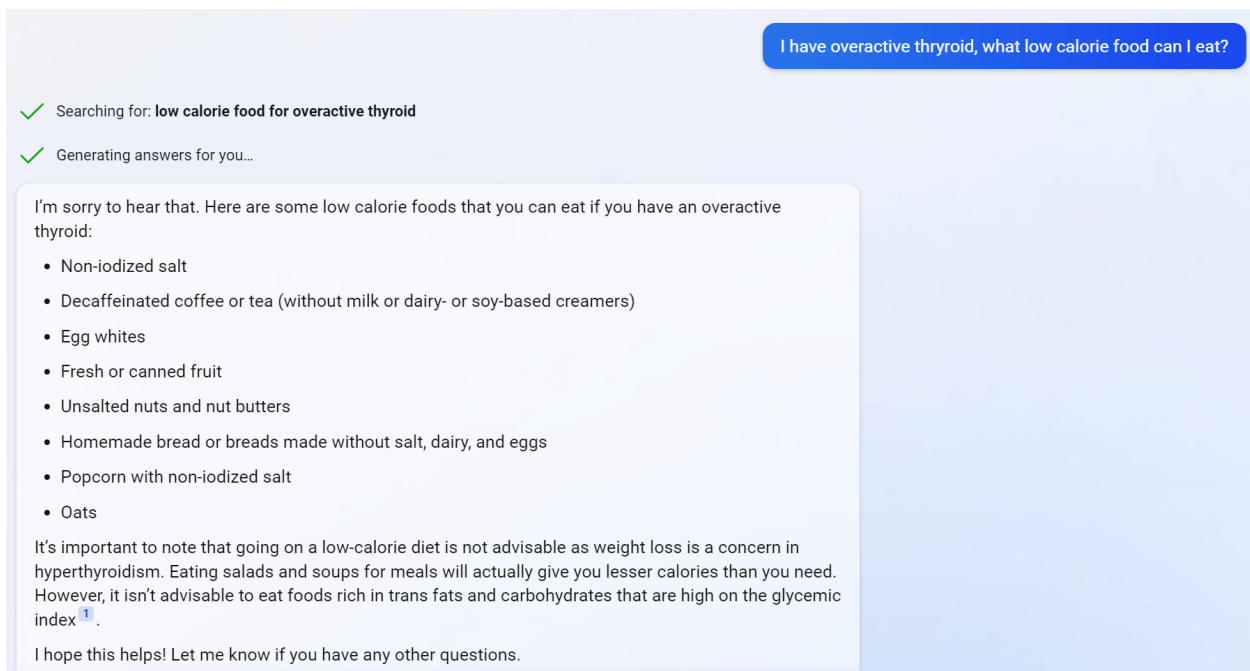
DD I have overactive thyroid, what low calorie food can I eat?



If you have an overactive thyroid, also known as hyperthyroidism, it's important to maintain a balanced diet that supports your overall health and provides adequate nutrition. While there isn't a specific low-calorie diet for hyperthyroidism, you may want to focus on nutrient-dense foods that are lower in calories. Here are some suggestions:

1. Lean proteins: Choose lean sources of protein such as skinless chicken breast, turkey, fish, tofu, and legumes. These foods are generally lower in calories and provide essential amino acids for muscle maintenance.
2. Non-starchy vegetables: Include a variety of colorful non-starchy vegetables in your diet, such as leafy greens, broccoli, cauliflower, bell peppers, zucchini, and mushrooms. These vegetables are low in calories and rich in vitamins, minerals, and fiber.
3. Whole grains: Opt for whole grains like quinoa, brown rice, oats, and whole wheat bread, which provide more fiber and nutrients compared to refined grains.
4. Healthy fats: Include sources of healthy fats such as avocados, nuts, seeds, and olive oil in moderation. These fats provide essential fatty acids and can help you feel satisfied.
5. Low-fat dairy or dairy alternatives: Choose low-fat or non-fat options for milk, yogurt, and cheese. If you are lactose intolerant or prefer dairy alternatives, go for unsweetened almond milk, soy milk, or other plant-based alternatives fortified with calcium and vitamin D.
6. Fruits: Enjoy a variety of fruits, such as berries, apples, oranges, and melons. They are generally low in calories and provide natural sweetness along with vitamins and minerals.

Figure 1.4: A large language model is unable to filter items for hyperthyroidism. These results are from ChatGPT⁴ on May 8, 2023



I have overactive thyroid, what low calorie food can I eat?

✓ Searching for: **low calorie food for overactive thyroid**

✓ Generating answers for you...

I'm sorry to hear that. Here are some low calorie foods that you can eat if you have an overactive thyroid:

- Non-iodized salt
- Decaffeinated coffee or tea (without milk or dairy- or soy-based creamers)
- Egg whites
- Fresh or canned fruit
- Unsalted nuts and nut butters
- Homemade bread or breads made without salt, dairy, and eggs
- Popcorn with non-iodized salt
- Oats

It's important to note that going on a low-calorie diet is not advisable as weight loss is a concern in hyperthyroidism. Eating salads and soups for meals will actually give you lesser calories than you need. However, it isn't advisable to eat foods rich in trans fats and carbohydrates that are high on the glycemic index ¹.

I hope this helps! Let me know if you have any other questions.

Figure 1.5: Retrieval augmented language models are unable to perform the right reasoning. These results are from Bing's chat feature on May 8, 2023

Subject	Relation	Object
hyperthyroidism	recommended_food	non-iodized salt
hyperthyroidism	recommended_food	unsalted nuts and nut butters
hyperthyroidism	recommended_food	egg whites
egg whites	calorie	low
unsalted nuts and nut butters	calorie	high
non-iodized salt	calorie	low
egg whites	is_a	food
unsalted nuts and nut butters	is_a	food
non-iodized salt	is_a	food

Table 1.1: Example questions and answers from the drop dataset, showing the relevant parts of the associated passage and the reasoning required to answer the question.

in a structured form. Let us say we have a semi-structured table containing the facts required to answer the question “I have an overactive thyroid, what low calorie food can I eat?” as shown in Table 1.1. If we adopt a semantic parsing [43] approach, then we need to learn to map natural language questions to an executable program like Eq. 1.1, which can be executed with a deterministic interpreter [44].

$$\begin{aligned}
& \lambda x.\lambda y \text{ Subject}(x, \text{'hyperthyroidism'}) \wedge \text{Relation}(x, \text{'recommended_food'}) \\
& \wedge \text{Relation}(y, \text{'calorie'}) \wedge \text{Object}(y, \text{'low'}) \wedge \text{Object}(x) == \text{Subject}(y)
\end{aligned} \tag{1.1}$$

When both the question and context can be parsed easily and effectively, the execution of reasoning follows a deterministic path. However, the task of semantic parsing remains formidable because parsing natural language into a semi-structured format poses significant challenges due to the contextual and inherently ambiguous nature of language itself [45]. Furthermore, models struggle to generalize to novel compositions, grammatical structures, and table headers without any adaptations [46].

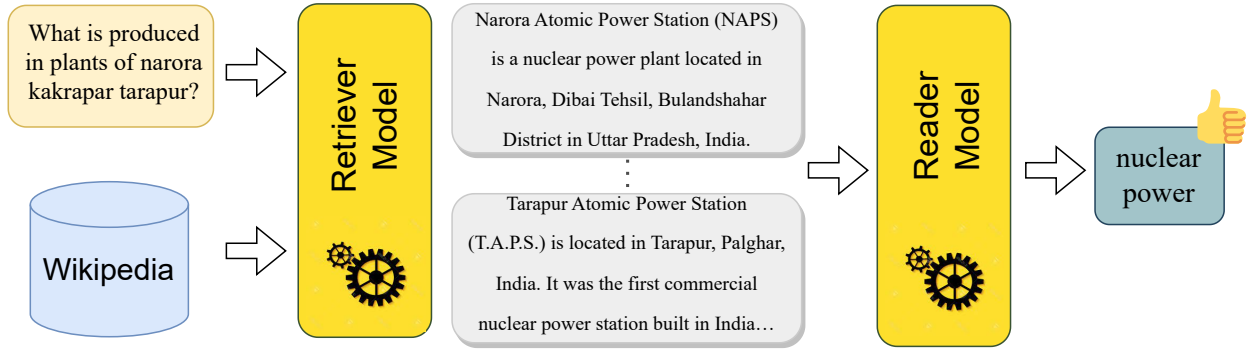
1.1.3 Generalizability

Neural systems are sensitive to changes in the distribution of data, especially when they are tested on examples that are very different from the original training distribution. For instance, in Figure 1.6a, we use a retrieve-and-read approach, where the retrieved top 100 passages from the Wikipedia document corpus are read by a reader model to answer the given question. If we extend the document corpus to also include Pubmed (biomedical articles), in an ideal scenario, the retrieved results and the final answer should not change. However, as illustrated in Figure 1.6b both retriever and reader models get confused and make an incorrect prediction. This is because the term “plant” in the biomedical corpus is often used to refer to the *living organism*, while in the retriever and reader models, which were trained on the Wikipedia corpus, they predominantly referred to the plant as a *location*.

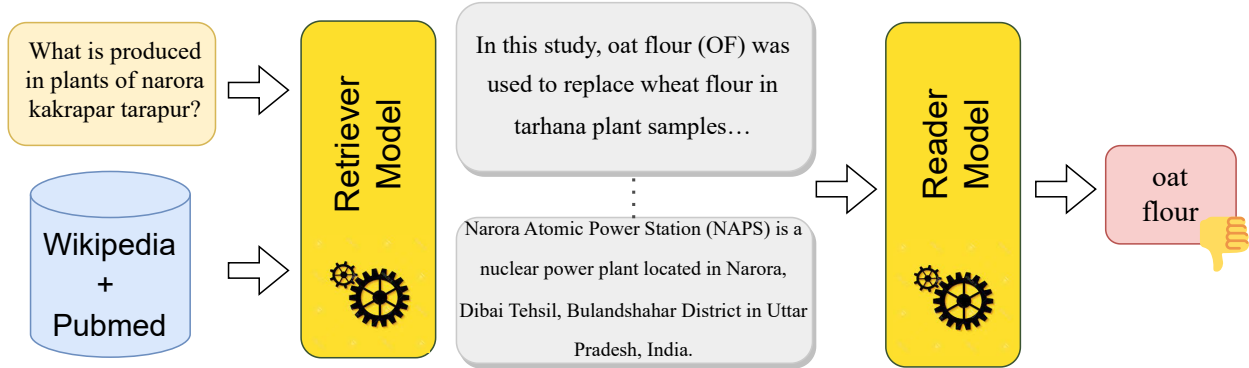
Both symbolic systems and neural systems face challenges in generalizing to unseen domains, but they suffer from different limitations. Symbolic systems require predefined logic rules in every domain to perform reasoning computations, and without these facts or rules, they fail to produce any output. On the other hand, neural systems tend to respond even when they encounter unfamiliar reasoning types that they were not trained on, as they lack the ability to introspect and accurately assess their own knowledge or lack thereof.

1.2 Background

Early question answering systems use rule-based bag-of-words [22, 47] style, shallow lexical matching methods to extract the span in a given document as an answer. These systems perform syntactic operations like coreference (pronoun) resolution and matching the wh-form of the question with the type of answer; for instance, questions that start with “Where” are associated with locations as answers. However, these systems are limited and difficult to generalize beyond the manually



(a) Both retriever and reader are able to identify the correct semantics associated with the word “plant”.



(b) Retriever gets perturbed and returns passages about plants further confusing the reader.

Figure 1.6: Retrieve-and-Read models trained on general-purpose domain with Wikipedia corpus and tested with Wikipedia and Pubmed corpus.

crafted rules. With the resurgence of neural networks [48, 49, 50] and advancements in large-scale dataset collection [51], it became easier to build QA systems that can learn rules and features from data directly without manual intervention. These systems do not rely on cumbersome and noisy linguistic features like dependency parsing, coreference resolution, semantic role labeling, etc., making them easier to deploy and use.

1.2.1 Notation

The question answering task can be formulated as a supervised prediction task where the goal is to predict the answer a for a given question q . To be able to answer the question, we also need supporting documents that contain references to the answer. When the set of supporting documents,

c_q are provided along with the question (q), it is referred to as *reading comprehension*. The goal of reading comprehension is to learn a model that predicts the answer given the question, q , and the relevant contexts, c_q , as $\hat{a} \leftarrow p(a|q, c_q)$. In some cases, we first need to retrieve the relevant documents c_q for a given question, a , and then answer the question based on these retrieved passages. This is referred to as *open-domain* question answering.

1.2.2 Foundation Models

With the success of plug-n-play object representations [52, 53, 34, 54] in computer vision, a series of works in natural language data sought to learn continuous representations for words in the vocabulary. The early works on learning continuous representations or embeddings for each word utilized word co-occurrences in the raw text to induce semantic meaning in the word representations. These methods [55, 56] learned a fixed embedding or representation for each word or token in the vocabulary. However, this approach was inadequate for representing natural language because the meaning of a word can vary widely depending on the context in which it is used. For instance, the word “bank” has different meanings in different contexts, like a food bank, river bank, blood bank, and a financial institution where we store money, etc. To address this issue, foundation models [57] were developed to learn contextual word representations [32, 58]. These models are initially trained or *pre-trained* on a large corpus of text to capture linguistic associations and features [59] that capture the meaning of words as used in different contexts in the natural language. Then, they are further trained or *fine-tuned* to solve a single or set of tasks like question answering, textual entailment, sentiment analysis, etc.

The fundamental building blocks of these models, known as ***Transformer*** blocks are shown in Figure 1.7. The two main components of a transformer block are the *multi-head attention* (MHA) and *feed-forward* network (FFN).

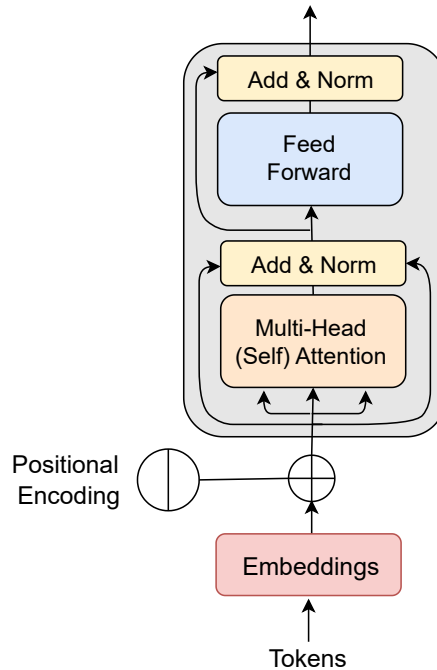


Figure 1.7: Transformer Block [1]

Token Representation: The first step is to map the raw text into a continuous space. A naive way to do this is to tokenize the text into words and then initialize each word with a random continuous vector which can be learned during training. However, Zipf’s law indicates that this leads to a large vocabulary size, which can be expensive to train and difficult to maintain – as new words are added to the natural language every day. On the other hand, based on morphological studies, words like “snowboard” and “skateboard” can be broken down into {“snow”, “board”} and {“skate”, “board”} thereby reducing the total number of words in the vocabulary. With these inspirations in mind, methods like byte-pair encoding [60] and word-piece [61] tokenization learn how to segment a piece of text to fit a fixed vocabulary size based on the frequency of continuous character segments in the text.

The input tokens, pertaining to a piece of text, are then mapped into continuous representations via the embeddings block as $X \in \mathbb{R}^{|\text{tokens}| \times d}$. One point to note is that the entire transformer block computes representations for all tokens in the input in parallel to promote faster computation. However, word ordering is crucial in natural language to understand the context. To represent order,

positional encodings are added to the input representations that specify the position of each token.

Multi-Head attention: The MHA block performs multiple parallel scaled dot-product attention. The dot-product attention is inspired by the lookup process in databases, where a query matches keys to retrieve corresponding values that are relevant to solving the query. To emulate a similar idea in a continuous space, three parameters: $W_Q \in \mathbb{R}^{d \times d}$, $W_K \in \mathbb{R}^{d \times d}$ and $W_V \in \mathbb{R}^{d \times d}$ are first randomly initialized. These parameters or weights learn to capture associations between various feature dimensions of the input during the course of training. A linear transformation maps the input into the same representation space as the above parameter to obtain an input-specific representation for Q (i.e, XW_Q), K (i.e, XW_K) and, V (i.e, XW_V).

The dot-product attention, computed as $QK^T \in \mathbb{R}^{|\text{tokens}| \times |\text{tokens}|}$, determines the correlation between the query vectors and the key vectors. However, as the dimensionality (d) of the parameters (i.e., W_Q, W_K, W_V) increases, the values resulting from QK^T also increase in magnitude. To prevent gradient explosion during model training, the dot-product is scaled down by \sqrt{d} .

$$\text{Attn}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d}} \right) V \quad (1.2)$$

The softmax function evaluates the significance of all neighboring tokens relative to each token in the input sequence. Then the value (V) representations are attenuated based on the overall importance of each token. A single head of attention (Eq. 1.2), is replicated and computed in parallel for a predefined number of times (say N) to obtain multi-headed attention – which is also referred to as *self-attention*. The outputs from each attention head are concatenated at each token position, as $[\text{Attn}(Q_1, K_1, V_1); \text{Attn}(Q_2, K_2, V_2); \dots ; \text{Attn}(Q_N, K_N, V_N)]$.

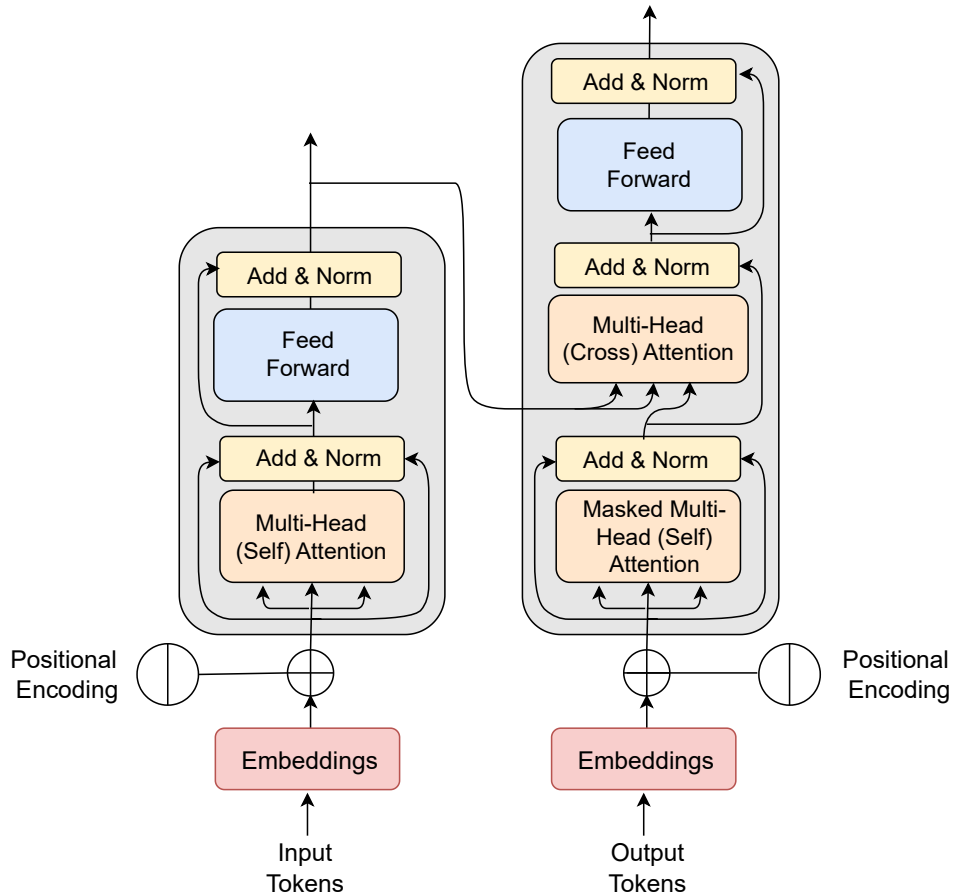


Figure 1.8: Encoder-Decoder Transformer Architecture

Feed-Forward: The output of MHA is then fed into FFN, with a receptive field of $d \times N$. The FFN block carries out a linear projection over representations at each token position in the input sequence. An FFN with a larger receptive field has a greater capacity to memorize knowledge facts [62].

The resulting output from the FFN serves as the encoded representation for individual tokens, which is then employed in the learning process for a given task. These representations can be pooled together to obtain a representation at the sentence level, which is useful for tasks such as sentiment analysis. Alternatively, they can be utilized for token-level classification tasks like part-of-speech tagging or even for answering long-form questions like “How to write a diary?”.

Encoder-decoder: In the case of long-form answer generation, the goal is to generate a sequence of tokens using the encoded representations of tokens in the input (question and context). This is achieved by generating tokens in an autoregressive manner, one step at a time, with well-known encoder-decoder [63] architecture. These models take a sequence as input to output another sequence, which is very useful in tasks like machine translation.

To maintain compatibility, the transformer blocks are used in both the encoder and the decoder. However, two modifications are applied in the case of the decoder. First, when learning to predict the token at each step, the self-attention is only computed across tokens at positions preceding the current step. This is because of the left-to-right nature of writing systems like Latin languages, where the subsequent word is composed based on knowledge of past tokens, not future tokens. A token-wise mask, M ensures access to only tokens on the left of the current token position by setting the softmax output of tokens on the right to zero.

$$\text{Masked-Attn}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d}} \odot M \right) V \quad (1.3)$$

Second, to capture associations between the decoded and encoded token representations (Figure 1.8), *cross-attention* is computed with query representation from the decoder and key and value representations from an encoder for each head and at each layer, as $\text{Attn}(Q^d, K^e, V^e)$, as shown in Figure 1.8

The number of associations or features learned by the overall model is determined by the number of parameters it contains. The number of parameters can be adjusted by stacking transformer blocks sequentially with different random initializations. One of the biggest problems with earlier models like Recurrent Neural Networks (RNNs) and Long-short term memory (LSTMs) was that it was unstable to train them over a long context and large number of parameters. However, with

gradient stability brought on by multi-head self-attention, layer normalization [64] and residual connections [65], the transformer architecture is capable of processing longer input sequences without encountering issues like vanishing gradient [66] in RNNs or gradient explosion [67] in LSTMs. Furthermore, the parallel nature of MHA and FFN computations, as opposed to the sequential operations in recurrent networks, has simplified the scaling of model parameters. This scalability is facilitated by hardware that is well-suited to accommodate such increased computational requirements. As a result, it has driven progress not only in natural language tasks but also in the realm of speech [68], videos [69] and images [69, 70]. Infact, the transformer architecture being a foundational backbone for represnetation learning across many modalities has led to progress in multi-modal [71] research.

In this dissertation, whenever we use the term “model”, we refer to a specific transformer model architecture that varies in the number of attention heads, transformer blocks (or layers), and the type of training losses used.

1.2.3 Question Answering Models

There are predominantly three ways to build question answering models: span-selection, conditional generation, and in-context prompting.

Span-selection This approach involves extracting text snippets from the given document as the answer. This model is trained in two stages: pre-training and fine-tuning. The pre-trained model, known as BERT [32], is trained with cloze task [72, 73] as the objective for learning contextual representations. Cloze task employs fill-in-the-blank problems as a way to learn context-sensitive word associations. To learn these representations, the BERT model masks a random word in a sentence and then predicts the word from a fixed vocabulary using the representation at the masked location. This allows the masked or predicted word to understand the context surrounding it.

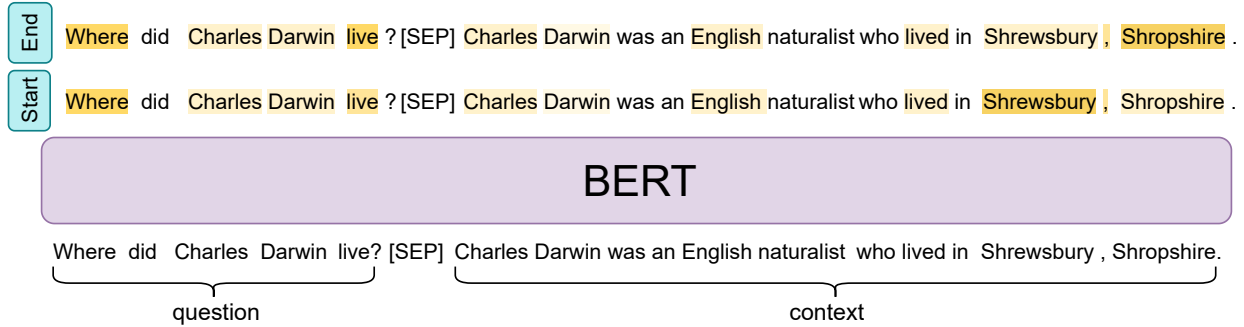


Figure 1.9: Question Answering as span selection problem: Probability distribution of start and end of answer spans, as heat map over words in the passage.

The fine-tuning step trains two classifiers – one to learn the probability that a token, t , in the given context(s) is the starting index of the answer span and another to learn if it is the ending index for the answer span. For instance, in Figure 1.9, the answer span start and end prediction classifiers assign a significant probability to the words “Shrewsbury” and “Shropshire” respectively, based on the given question and context. During inference, the probability of all spans, starting at each index up to a predefined fixed length, l , is calculated and then ranked to obtain the answer span with the highest probability, i.e. “Shrewsbury, Shropshire”.

$$(\hat{s}, \hat{e}) \leftarrow \underset{s \in [0, |\text{tokens}|], e \in [s, s+l]}{\text{argmax}} \log p_{\text{start}}(t_s | q, c_q) + \log p_{\text{end}}(t_e | q, c_q) \quad (1.4)$$

$$\hat{a} = \text{tokens}[\hat{s} : \hat{e}] \quad (1.5)$$

Conditional Generation This approach involves generating the answer given a question and a document. This modeling style is based on encoder-decoder models that map an input sequence to an output sequence. The advantage of such a training approach is that it standardizes the format for various tasks, making it easier to multi-task since any task can be redefined as an input-output sequence. For instance, UnifiedQA [74] unifies question answering tasks with different

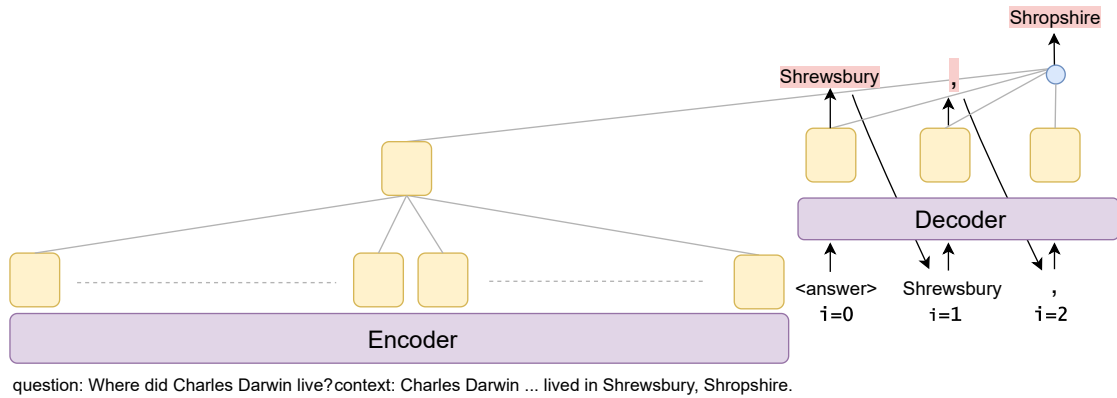


Figure 1.10: Question Answering as generation problem: At each step (i), the next token is generated based on representations from input tokens and previously generated tokens

answer formats like multiple-choice [75], span-selection [76], boolean (yes/no) [77] and free form answer [78]. In fact, studies [58, 79] have demonstrated that training on multiple tasks concurrently leads to improved transfer of linguistic understanding across various tasks and even results in the emergence of new abilities that were not observed during training.

Similar to span selection, we first need to pre-train the model on a large amount of unstructured natural language data with a cloze-task to impart linguistic knowledge. Since this is now a generation model rather than a classification model, we use a placeholder token (a.k.a sentinel) that does not exist in the vocabulary to replace the word at the masked location. The model is then prompted with this same placeholder token to generate the correct word at the masked location. At the fine-tuning stage, the model is prompted with a placeholder token, like “<answer>” in Figure 1.10 to generate answer tokens in the subsequent steps until the end of the sequence token is generated.

$$\hat{a}_i \leftarrow p(a_i | a_{<i}, q, C) \tag{1.6}$$

Incontext prompting In this approach, a large language model is provided with instructions along with a few examples that demonstrate how to solve a specific task. Subsequently, the model

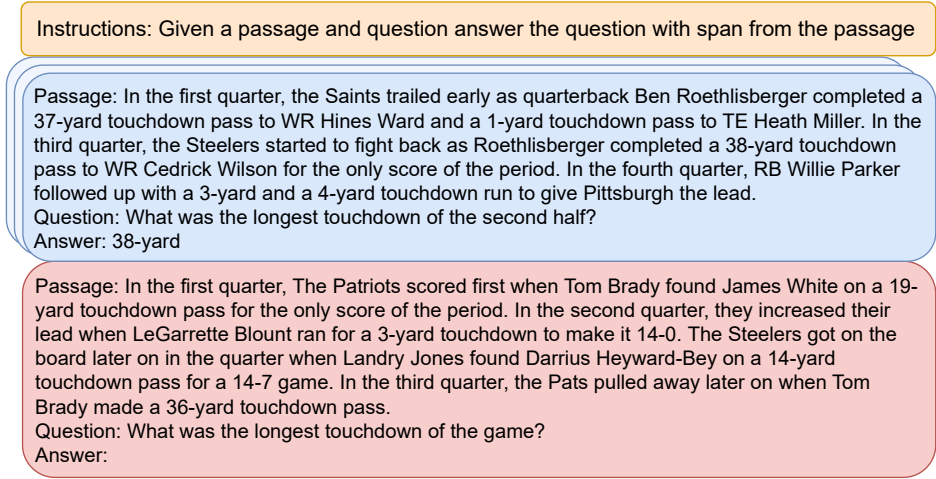


Figure 1.11: The instructions explain the task to be performed in natural language, and then examples (in blue) demonstrate how to solve examples analogous to the test example (in pink). The model is prompted with “Answer:” to continue producing the answer to the given question.

is prompted to solve a test question based on the given instructions and examples as shown in Figure 1.11. Prompting [42] has recently transformed the NLP methodologies by shifting away from training individual models for each task, such as question answering, machine transition, or summarization. Instead, a single large language model is pre-trained auto-regressively on a vast language corpus to generate the next token and subsequently fine-tuned on numerous instruction-following tasks [80, 81, 82]. This yields a general-purpose model that is capable of solving a wide range of new tasks by solely looking at instructions and example demonstrations in the prompt during test time.

1.2.4 Forms of Reasoning in Question Answering Datasets

Simple Reasoning This style of reasoning entails conducting surface-level matching to search for text spans that provide an answer to a question [76]. For instance, as shown in Fig. 1.12, to answer the question, **Which player made the 3-yard touchdown**, a neural QA model performs lexical matching with the span, **LaGarette Blount ran for a 3-yard touchdown** and then extracts the specific span that matches the wh-form “Which player” to extract player name “LaGarette Blount” as the

In the first quarter, The Patriots scored first when Tom Brady found James White on a 19-yard touchdown pass for the only score of the period. In the second quarter, they increased their lead when LeGarrette Blount ran for a 3-yard touchdown to make it 14-0. The Steelers got on the board later on in the quarter when Landry Jones found Darrius Heyward-Bey on a 14-yard touchdown pass for a 14-7 game. In the third quarter, the Pats pulled away later on when Tom Brady made a 36-yard touchdown pass.

Q: In which quarter did LeGarrette make the 3-yard touchdown?

A: second

Q: Which player made the 3-yard touchdown?

A: second

Q: What was the longest touchdown of the game?

A: 36-yard

Figure 1.12: Reasoning types in QA datasets: The first two are simple questions while the last question is compositional

answer.

Compositional Reasoning This style of reasoning entails performing compositional reasoning over multiple simple facts in the passage [17]. For instance, to answer the question, What was the largest touchdown of the game?, the model needs to locate all the touchdowns in the game and then perform a max operation on them to find the largest value.

Other Reasoning There are other reasoning types that require commonsense reasoning [83], integrating world knowledge [84], understanding belief state of various character [85], temporal reasoning [86] and spatial reasoning [87] etc. but we would not be focusing on them in this work.

1.3 Dissertation Outline

Thesis Statement: The deficiencies in reasoning abilities of neural question answering systems, namely consistency, decomposability, and generalizability, can be overcome through targeted data interventions and training paradigms.

In particular, we make three main contributions:

- We demonstrate that neural models exhibit unpredictable behavior when operating under the independence assumption. We propose a way to generate related QA pairs and show that incorporating joint training of question and answer pairs, and even training with multiple question and answer pairs, can effectively promote consistency in the models' performance.
- We introduce a new benchmark dataset that tests the compositional reasoning ability of QA systems and shows that purely neural systems struggle on this task. To further promote decomposability, we introduce a novel approach, "Successive Prompting", that uses data augmentations to break down a complex problem into simpler sub-problems. This enables the model to leverage shared sub-problems across compositional questions, even when they are not similar on the surface.
- We propose a test to assess whether a neural QA model will effectively generalize to a new domain without any specific adaptation in the new domain. We further show the effectiveness of three different kinds of interventions: (1) zero and few-shot data augmentation, (2) sampling methods, and (3) intermediate annotations detailing steps for compositional tasks, towards domain generalization.

Organization: This dissertation is divided into three parts. In the first part, we show that independence assumptions and bias in training data causes inconsistencies in model learning and we propose two methods to alleviate this problem. In the second part, we develop an evaluation

benchmark that shows neural systems struggle with compositional reasoning. We, further, propose a method to decompose compositional questions into simpler questions which exploits shared simple or sub-questions as supervision signals for improving few-shot performance. In the final part, we demonstrate conditions under which generalization fails and propose different types of interventions like sampling and data augmentation to improve generalization.

1.4 Key Findings

We present key findings across several works below.

1.4.1 Imposing Non-IID constraints improves consistency while answering.

Neural systems are stochastic in nature, which can produce inconsistent results when tested with minimally different examples [88, 89], as discussed in Figure 1.2. Our hypothesis is that this occurs because of the independence assumption while data collection and training of neural models. To test this hypothesis, we propose a method to generate bundles of related QA pairs. We then impose bijective constraints on these *Instance Bundles* such that each question is aligned with a unique answer. By imposing these constraints at inference alone, the model performance trained with conventional maximum likelihood estimation (MLE) loss improves performance by 6.5%. Additionally, by incorporating contrastive training (CE), which jointly learns to answer all related questions in the bundle, we further improve the performance, reaching upto 90.1%.

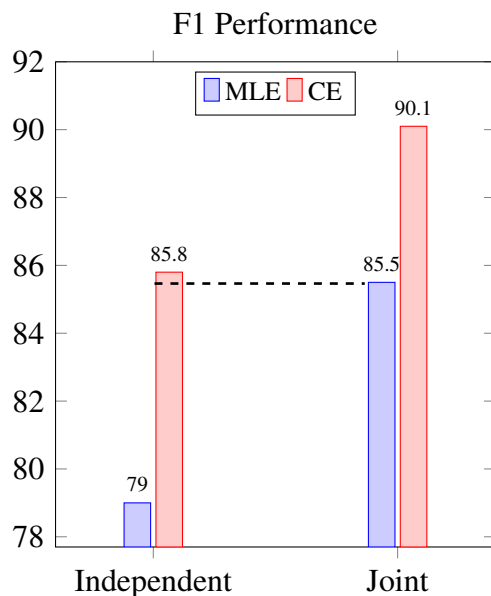


Figure 1.13: Performance (F1) on HotpotQA dev set with IID vs non-IID inference for a model trained with MLE vs CE.

1.4.2 Generative QA improves consistency while passage retrieval.

Neural systems are adept in capturing statistical correlations within data, but this also implies that they can inadvertently learn and perpetuate biases that exist within the data distribution [90, 13, 91, 92]. Passage retrieval in multi-hop QA is also strife with bias in data and relies on shallow lexical cues leading to incorrect reasoning [2]. We hypothesize that this is because the discriminative training for passage selection optimizes the 0/1 loss, which solely considers whether the oracle passages are ranked higher or not. As a result, it does not incentivize paying attention to the conditioning variable (i.e., question) in its entirety, but enough to solve the dataset – which further exacerbates bias in model learning.

We propose a *generative model* where the passage selector is trained to generate the question given the context. This approach forces the retriever not to overlook any aspects of the question. Along with the conditional question generation model, we learn a prior over pair of passages that capture the compatibility between passages, which is especially useful for multi-hop question

answering. Figure 1.14 shows that the performance of the generative retriever is better by $\sim 5\%$ over the discriminative retriever on an adversarial held-out set.

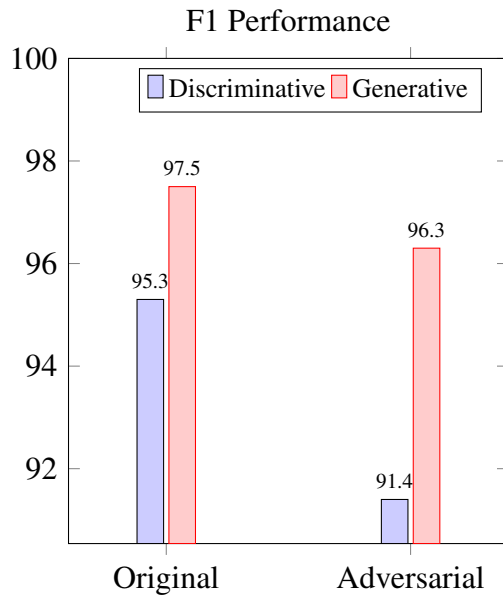


Figure 1.14: Performance (Acc) comparison of discriminative vs generative model with original and adversarial dev set.

1.4.3 Purely neural systems lack Compositional reasoning

In spite of having reached human-level performance in solving simple questions (Figure 1.15), can advancements in neural architectures for simple reasoning be applied to more complex forms of reasoning? To answer this question, we introduce a new dataset DROP that requires composing multiple facts to solve complex problems. In Figure 1.16, we show performance of a series of models including symbolic, neural and hybrid (or neuro-symbolic) models on the DROP dev set. Symbolic systems need to first parse natural language questions into an executable expression which can be directly executed [93] and context into a semi-structured format on which the said expression can be executed. We observe that even state-of-the-art [94] systems for parsing passages struggle and produce a very low yield, resulting in low performance.

Purely neural models [95] trained on DROP (Neural-FT) do not perform as well as neuro-symbolic

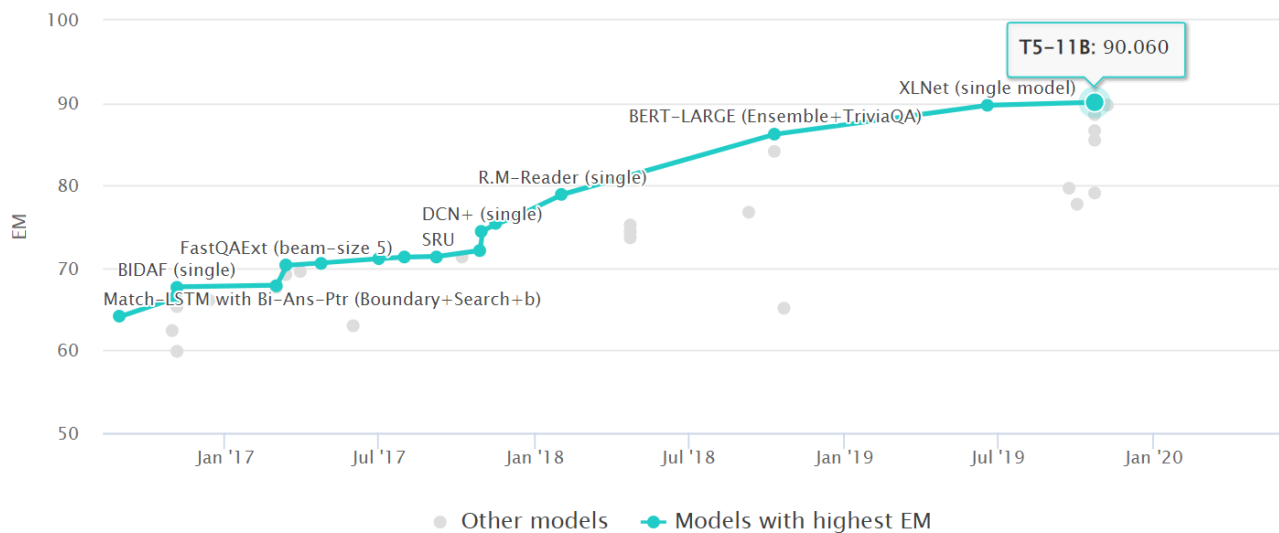


Figure 1.15: Successive prompting exploits shared sub-problems from questions that are not similar on the surface

(Hybrid) systems. The neuro-symbolic approach combines neural architecture for feature extraction, specifically addressing the limitations of Symbolic systems, with symbolic operations for performing the final computations. Even large language models [96, 97] that are considered general-purpose and trained on trillions of tokens with parameters in the order of hundreds of billions (Neural-LM) are not able to match up to neuro-symbolic models – which are trained with much less data and have fewer than a billion parameters.

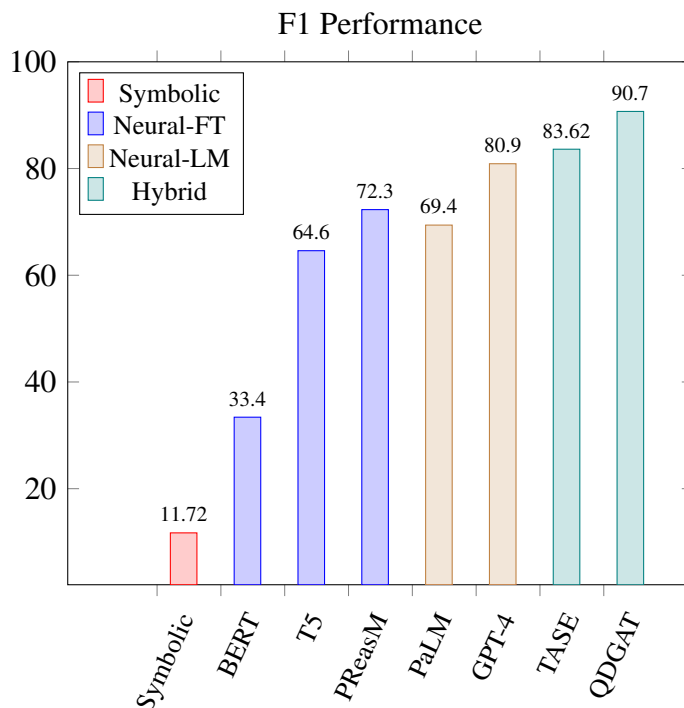


Figure 1.16: Purely Symbolic and Neural systems do not perform as well as Neuro-Symbolic (Hybrid) systems.

1.4.4 Decomposing complex problem improves few shot model performance

Chain-of-Thought [98] has become a popular way to prompt an LLM with instructions and a few examples of how to solve a task at inference time alone. This forces the model to reason about the intermediate steps with aid from similar complex questions and demonstrations on how to solve them. However, similar examples are looked up based solely on the complex question to be solved at hand, which fails to fully leverage all potential examples that may share a sub-problem despite their surface-level differences.

We propose *Successive prompting* that breaks down a complex problem into simpler problems and successively looks up appropriate demonstrations on how to solve the simple problem. For instance, in Figure 1.17, we observe that the test question τ = “Which player made the longest touchdown of the game?” on the surface is closer to “Which player made the longest touchdown in the first

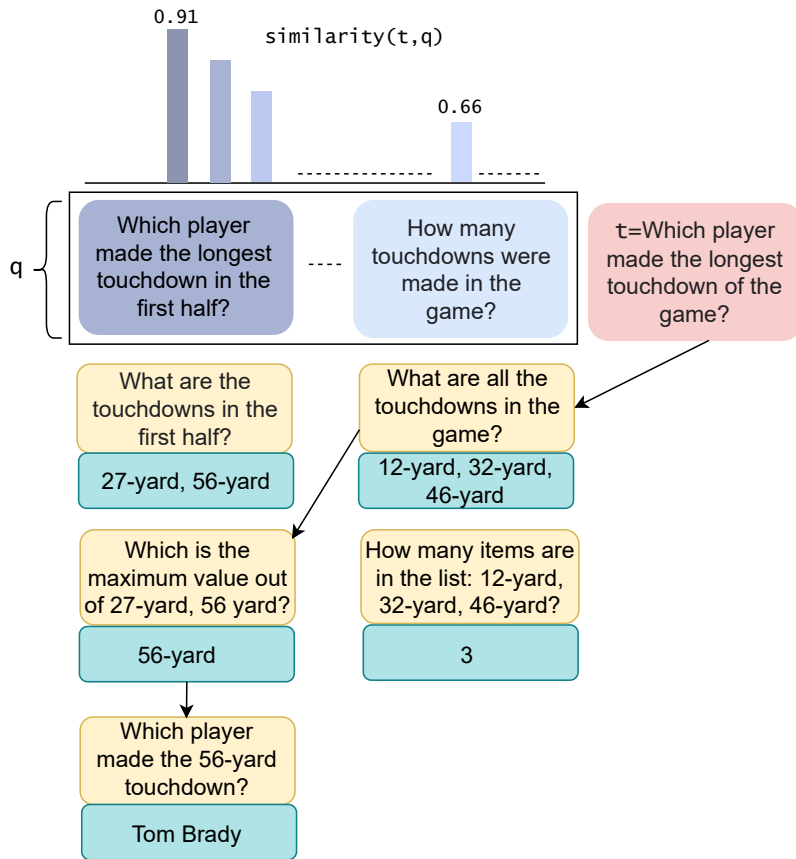


Figure 1.17: Successive prompting exploits shared sub-problems from questions that are not similar on surface

half?” and is more likely to get selected as a similar example for demonstration over “How many touchdowns were made in the game?” However, the latter question provides much better supervision for solving the first step. Overall, by successively querying for demonstrations at each step, we retrieve the most relevant examples for solving that step. In Figure 1.18, we show that our method quantitatively improves over Chain-of-Thought by ~4%.

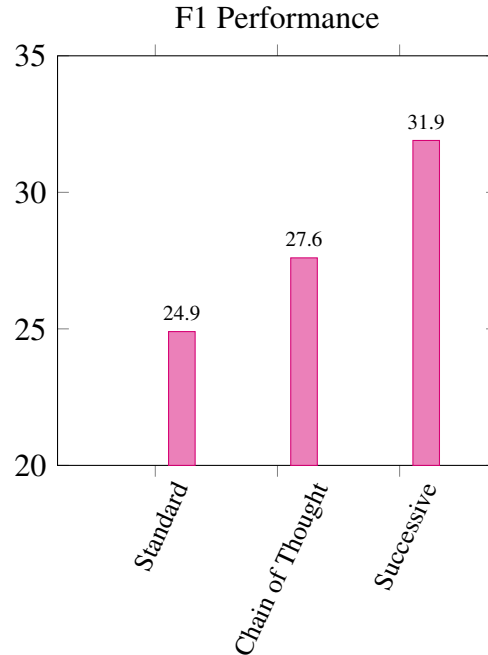


Figure 1.18: Decomposing complex problems into simpler problems allows better generalization.

1.4.5 Type of dataset shift governs the effectiveness of data augmentation strategies.

While neural QA models excel at learning to solve a specific dataset, their learnings may not extend to different datasets out of the box. Determining the generalizability of a model to new domains often requires collecting a reasonable number of samples to train an effective in-domain model. We introduce a generalizability test that can characterize the type of dataset shift in the new dataset with respect to an existing QA model (source) – using only a limited number of test examples from the new domain. In Figure 1.19, we show that the average end-to-end performance of the source model is quite poor when applied to new datasets (on the top) but improves when adapted to unseen domains with augmentation strategies. We then drill down (below) into the effectiveness of zero and few-shot data augmentation strategies and show that the datasets exhibiting covariate and concept shift respond to both zero and few-shot data augmentations. However, datasets without any shift (i.e., close to source domain) do not improve much with any augmentation, while datasets with Full

shift benefit the most from few-shot adaptation, as it aligns more closely with its data distribution.

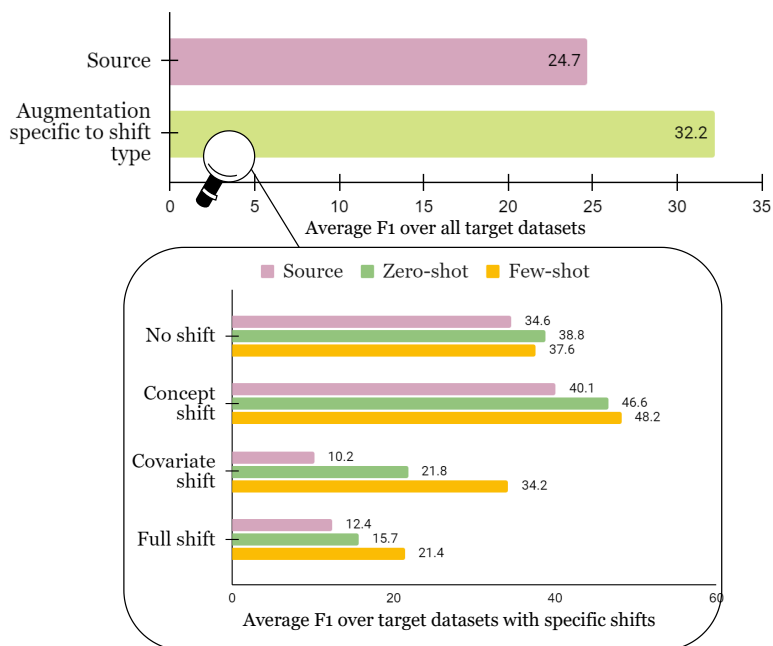


Figure 1.19: Effect of data augmentation techniques on dataset shifts

1.4.6 Sampling data proportional to increase in loss improves generalization

Multi-task learning [99] is often marred by underfitting on high resource tasks and overfitting on low resource tasks [8, 14]. This is because it is often difficult to balance the sample representation of each dataset via a pre-defined fixed sampling schedule. Our hypothesis is that drop in performance on a specific dataset is due to catastrophic forgetting [100]. To mitigate this issue, we propose *Dynamic Sampling*, which samples more examples from the dataset that the model has forgotten how to handle effectively, despite having previously acquired the knowledge to do so. We do this by sampling examples in proportion to the drop in the performance on the held-out set for each dataset over the epochs.

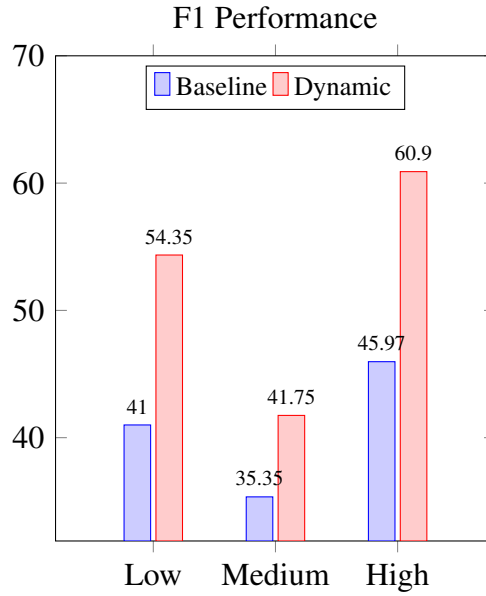
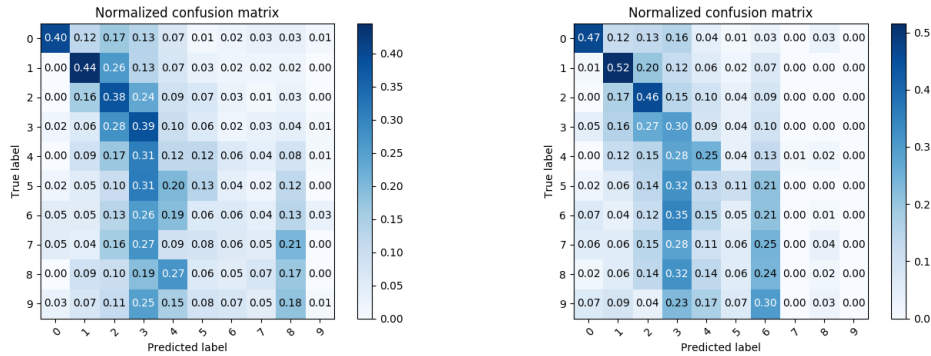


Figure 1.20: Baseline vs. Dynamic sampling averaged over low, medium and high resource datasets in ORB

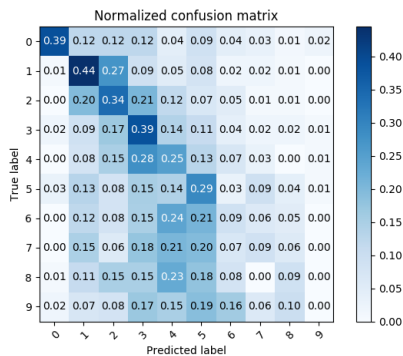
1.4.7 Intermediate annotations reduce bias and improve generalization

To solve compositional problems, QA models need to identify latent sequential operations and execute them only with supervision from the final answer alone. The lack of supervision for intermediate steps can result in learning unanticipated biases in the data. We demonstrate that supervising the model with annotations that emphasize the relevant sections of passages for answering questions, alongside the question answering objective, aids in mitigating label bias – while supervising with just more QA pairs exacerbates the problem. For instance, in Figure 1.21, we show the confusion matrix over number-type answers that range from $[0 - 9]$ in the dev set. A model initially trained on number focused subset (10k samples) of DROP is biased towards numbers $\{0, 1, 2, 3\}$. The same model, when subsequently fine-tuned with additional QA pairs, gets more biased around the number “3”. However, when the model is subsequently fine-tuned with intermediate annotations on existing examples, the predictions get more diffused and less biased Figure 1.21.



(a) 10k samples

(b) Additional QA pairs



(c) Intermediate Annotations

Figure 1.21: For the same cost, intermediate annotations help diffuse biased over-representation of number 3 as compared to adding more question-answer pairs

Published Work and Collaborations This dissertation is based on the following published works.

- Our work on Instance Bundles (Chapter 2), which jointly learns to answer closely related QA pairs, was published in 2021 Conference on Empirical Methods in Natural Language Processing with collaborators: Pradeep Dasigi, Sameer Singh, Matt Gardner
- The work pertaining to the generative conditional QA model that improves the consistency of context selection (Chapter 3) was published in 2021 Conference on Empirical Methods in Natural Language Processing in collaboration with Cicero Santos, Patrick Ng, Ben Athiwaratkun, Bing Xiang, Matt Gardner, and Sameer Singh.
- The DROP dataset, discussed in Chapter 4, for evaluating compositional reasoning in Neural QA systems was published in the 2019 Annual Conference of the North American Chapter of

the Association for Computational Linguistics in collaboration with Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh and Matt Gardner.

- The Successive Prompting method, discussed in Chapter 5, was published in 2022 Conference on Empirical Methods in Natural Language Processing in collaboration with Shivanshu Gupta, Sameer Singh and Matt Gardner.
- Chapter 6 delves into conditions under which generalization fails and explores various data interventions that can effectively improve generalization. This work was published in 2023 Conference on Association for Computational Linguistics in collaboration with Emma Strubell, Sameer Singh, Pat Verga.
- The ORB benchmark, in Chapter 7 was proposed to evaluate the performance of question answering models across a variety of reasoning types. This work was published in the 2nd Workshop on Machine Reading for Question Answering in collaboration with Ananth Gottumukkala, Alon Talmor, Matt Gardner and Sameer Singh.
- In Chapter 8, we propose Dynamic Sampling that alleviates catastrophic forgetting in multi-task setting. This work was done in collaboration with Ananth Gottumukkala, Sameer Singh and Matt Gardner and published at 2020 Conference on Association for Computational Linguistics.
- Finally, Chapter 9 shows that providing intermediate annotations for complex questions helps reduce bias in compositional reasoning. This work was published at 2020 Conference on Association for Computational Linguistics in collaboration with Sameer Singh and Matt Gardner.

Part I

Improving Consistency: Objectives for Joint Modeling

Chapter 2

Instance Bundles

Consistency is what transforms average
into excellence.

Linda Kimonyi

When training most modern reading comprehension models, all the questions associated with a context are treated as being independent from each other. However, closely related questions and their corresponding answers are not independent, and leveraging these relationships could provide a strong supervision signal to a model. Drawing on ideas from contrastive estimation, we introduce several new supervision techniques that compare question-answer scores across multiple related instances. Specifically, we normalize these scores across various neighborhoods of closely contrasting questions and/or answers, adding another cross entropy loss term that is used in addition to traditional maximum likelihood estimation. Our techniques require bundles of related question-answer pairs, which we can either mine from within existing data or create using various automated heuristics. We empirically demonstrate the effectiveness of training with instance bundles on two datasets—HotpotQA and ROPES—showing up to 11% absolute gains in accuracy. The emphasis on leveraging relationships and striving for consistency helps propels performance from being average

and even random in some cases.

2.1 Introduction

Machine learning models are typically trained with the assumption that the training instances sampled from some data distribution are independent and identically distributed. However, this assumption can cause the learner to ignore distinguishing cues [101] between related or minimally different questions associated with a given context, resulting in inconsistent model learning [102, 89]. In a dataset like ROPES, where the ideology of collecting pairs of minimally different questions is taken to its extreme, we see that the performance of a competitive baseline model (RoBERTA) is close to random [103]. One potential reason for this poor performance is that the model considers each question independently, instead of looking at differences between related questions.

Context: Marsilea is a genus of approximately 65 species of aquatic ferns of the family Marsileaceae. The name honours Italian naturalist Luigi Ferdinando Marsili (1656-1730) Brabejum is a genus of a single species of large evergreen tree, Brabejum stellatifolium in the Proteaceae, commonly called wild almond, bitter almond or ghoeboontjie.

Question 1: Is the Marsilea or the Brabejum the genus of more individual species of plants?

Answer 1: Marsilea

Question 2: Is the Marsilea or the Brabejum the genus of less individual species of plants?

Answer 2: Brabejum

Figure 2.1: Instance bundle created from HotpotQA

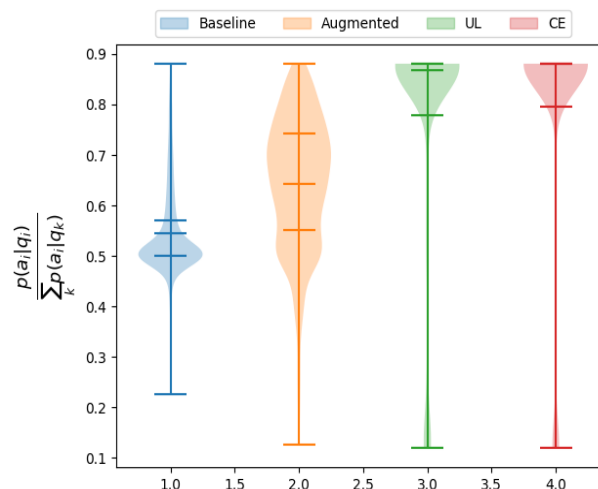


Figure 2.2: Probability of gold QA pair normalized over all questions in the bundle. The higher value indicates that positive QA pair has a high likelihood and at the same time negative QA pair has a low likelihood. At 0.5, both the contrastive questions would produce the same answer with the same likelihood. In an ideal scenario, the distribution should be a delta function at 1.0

To address this problem, we propose to train models with sets of related question-answer (QA) pairs simultaneously, instead of having a loss function that decomposes over independent examples. We use the term *instance bundle* to refer to these sets of closely contrasting examples. Consider an instance bundle from HotpotQA in Figure 2.1, containing two contrastive QA pairs, which differ in their input by only one word (changing *more* to *less*), resulting in different answers. With both of these examples in a training set, a model trained with traditional maximum likelihood estimation will be incentivized to figure out the difference between their inputs that leads to the expected difference between their answers, but the instances are likely to be seen far apart from each other during training, giving only a weak and indirect signal about their relationship.

In order to more effectively learn from these instance bundles, we draw on contrastive estimation [104], a method for re-normalizing an unsupervised probabilistic model using a neighborhood of related examples (originally a set of perturbations of some observed text). We extend this technique to apply to supervised reading comprehension problems by carefully selecting appropriate “neighborhoods” from instance bundles. The simplest choice of neighborhood is the set of contrasting answers from the instance bundle, resulting in a method similar to unlikelihood training [105] or noise-contrastive estimation [106]. However, there are other choices, including the set of contrasting *questions*, or combinations of questions and answers. These re-normalized loss functions are not effective on their own, which is likely why they have not been used before for training reading comprehension models, but when combined with maximum likelihood training they give substantial increases in performance.

An intuitive explanation of the reason for this performance improvement is shown in Figure 2.2. When trained on non-contrasting data with maximum likelihood estimation, a model gives roughly equal values for both $p(A_1|Q_1)$ and $p(A_1|Q_2)$, even though Q_1 and Q_2 are in some sense opposites. Adding the contrasting data helps the model differentiate these two probabilities, but not as much as unlikelihood training, which itself is not as effective as contrastive estimation.

We empirically demonstrate the utility of this approach on two reading comprehension datasets:

HotpotQA [40] and ROPES [103]. In order to use instance bundles on these datasets, we introduce various heuristics for obtaining closely related instances. We show that using contrastive estimation on the instance bundles that we obtain gives up to an 11% absolute performance improvement over prior training techniques. These results strongly suggest that data should be collected in instance bundles wherever possible, to allow for stronger supervision signals during training.

2.2 Contrastive Estimation for Reading Comprehension

Reading comprehension is the task of producing an answer a given a question q about a context c . The question is tied to a particular passage, so in the discussion that follows we will typically use q as a shorthand to refer to both q and c together. Reading comprehension models are typically trained to maximize the likelihood of the answer to each training question. Given a model’s exponentiated scoring function $\psi(q, a)$ for a QA pair,¹ this objective normalizes the scores over all possible answer candidates \mathcal{A} for a given question:

$$\begin{aligned} \mathcal{L}_{\text{MLE}}(q_i, a_i) &= \log p(a_i|q_i) \\ &= \log \frac{\psi(q_i, a_i)}{\sum_{c \in \mathcal{A}} \psi(q_i, c)} \end{aligned} \tag{2.1}$$

In this work we use a generative model for ψ , but many other alternatives are available, and our contribution is applicable to any scoring function. Specifically, we use as ψ the (locally normalized) probability assigned by the generative model to an answer candidate for a given question.

Instead of normalizing scores over all possible answer candidates, *contrastive estimation* [104] normalizes scores over some *neighborhood* of closely related instances. This method was originally introduced for unsupervised linguistic structure prediction, with a neighborhood obtained by permuting observed text to get inputs that had similar content but were ungrammatical. Our

¹ ψ is parameterized by model parameters θ , but we omit this in the equations for simplicity of exposition.

contribution is to apply this general idea to supervised reading comprehension problems. In our setting, given a neighborhood $\mathcal{N}(q, a)$ of related QA pairs, CE can be described as

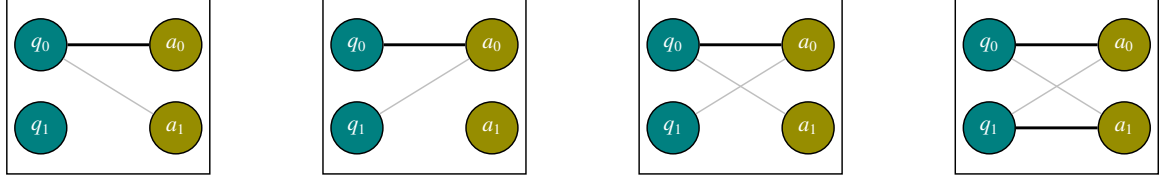
$$\mathcal{L}_{CE}(q_i, a_i) = \log \frac{\psi(q_i, a_i)}{\sum_{q_j, a_k \in \mathcal{N}(q_i, a_i)} \psi(q_j, a_k)} \quad (2.2)$$

[104] *replaced* the MLE objective with CE, which worked well in their unsupervised learning problem. In supervised learning, MLE is a much stronger training signal, and CE on its own severely underperforms MLE. This is because CE provides no learning signal for the very large space of alternative answers to a question that are not in the neighborhood. However, CE can provide a much stronger signal than MLE for a small set of potentially confusing alternatives, as there are fewer ways for the model to increase the probability of the correct answer. To adapt CE to supervised settings we interpolate between the two losses, instead of replacing MLE with CE:

$$\mathcal{L} = \alpha_1 \mathcal{L}_{MLE} + \alpha_2 \mathcal{L}_{CE} \quad (2.3)$$

Interestingly, this can be viewed as forcing the scoring function ψ to permit multiple different probabilistic interpretations, as both losses perform softmaxes over different sets of alternatives. Additionally, if ψ has some locally-normalized component, as is true for the generative models we work with and for many other common models (such as BIO tagging, or independent span start and span end positions), this interpolation in some sense trades off between the locally-focused MLE and the more global view of the problem that the normalization in CE provides (see §2.4.3 for further discussion of this point).

The key question in applying CE to reading comprehension is how to choose a neighborhood \mathcal{N} for a given training example. We do so by making bundles of related instances, then extracting various combinations of questions and answers from a bundle to use as neighborhood. Formally, an



(a) Answer Conditional (b) Question Conditional (c) Full partition (d) Joint

Figure 2.3: Contrastive Estimation models. In each subfigure an instance bundle of size 2 is shown, with bold lines indicating combinations whose probability is maximized at the expense of the combinations represented by gray lines, for the positive QA pair (q_0, a_0) in the bundle. The total CE loss is the sum of loss for each positive QA pair in the bundle.

instance bundle \mathcal{B} is a collection of unique questions \mathcal{B}_Q and unique answers $\mathcal{B}_\mathcal{A}$, such that there is at least one QA pair where a is the correct answer to q : $\text{ans}(q) = a$. We refer to such pairs as (q_g, a_g) in the discussion that follows. Our assumption is that the questions in \mathcal{B}_Q and the answers in $\mathcal{B}_\mathcal{A}$ are related to each other in some way—often they differ in only a single word—though we do not characterize this formally. However, a good bundle creation procedure is crucial for effective model learning. We discuss several ways for creating bundles in §2.3, and discuss the limitations of CE when effective bundles cannot be created in §2.5.2. The following section discusses choices of neighborhood functions given an instance bundle.

2.2.1 Choosing a neighborhood

Given an instance bundle \mathcal{B} with questions \mathcal{B}_Q and answers $\mathcal{B}_\mathcal{A}$, there are many ways to construct a neighborhood. Figure 2.3 shows some of these options graphically, with the bold line showing the gold QA pair, and gray lines showing the other QA pairs that make up the neighborhood. We distinguish between two kinds of neighborhood methods. A *single neighborhood* CE model is one that perturbs and normalizes over a single variable, either the question (input) or the answer (output). Similarly, *multiple neighborhood* CE models perturb both variables jointly and normalize over the combinatorial space of both variables.

Single neighborhood models

These models construct neighborhood using the set of contrasting answers or contrasting questions from the instance bundle.

Answer Conditional: This probabilistic model maximizes the probability of the correct answer a_i at the expense of the other answers candidates in the instance bundle $\mathcal{B}_{\mathcal{A}}$ (Figure 2.3a).

$$\mathcal{L}_{\text{CE-AC}}(q_g, a_g, \mathcal{B}) = \log \frac{\psi(q_g, a_g)}{\sum_{a_j \in \mathcal{B}_{\mathcal{A}}} \psi(q_g, a_j)}$$

Question Conditional This model computes the normalization constant over the question neighborhood for a fixed answer. This effectively computes a probability distribution over *questions* in the bundle given the correct answer, and maximizes the probability of the correct question (Figure 2.3b).

$$\mathcal{L}_{\text{CE-QC}}(q_g, a_g, \mathcal{B}) = \log \frac{\psi(q_g, a_g)}{\sum_{q_j \in \mathcal{B}_{\mathcal{Q}}} \psi(q_j, a_g)}$$

Multiple neighbourhood models

These models consider all possible combinations of questions, $\mathcal{B}_{\mathcal{Q}}$ and answers, $\mathcal{B}_{\mathcal{A}}$ in a bundle for normalization, unlike single neighborhood models which only look at either $\mathcal{B}_{\mathcal{A}}$ or $\mathcal{B}_{\mathcal{Q}}$.

Two Way This method simply does a weighted combination [107] of the answer conditional and question conditional losses.

$$\mathcal{L}_{\text{CE-TW}}(q_g, a_g, \mathcal{B}) = \lambda_1 \log \frac{\psi(q_g, a_g)}{\sum_{a_j \in \mathcal{B}_{\mathcal{A}}} \psi(q_g, a_j)} + \lambda_2 \log \frac{\psi(q_g, a_g)}{\sum_{q_j \in \mathcal{B}_{\mathcal{Q}}} \psi(q_j, a_g)}$$

Full Partition Instead of separate normalizations over questions and answers, this method does a single normalization over the same sets as in Two Way. This is equivalent to normalizing over the cross product $\mathcal{B}_Q \times \mathcal{B}_A$, minus other correct pairings (Figure 2.3c).

$$\mathcal{L}_{\text{CE-FP}}(q_g, a_g, \mathcal{B}) = \log \frac{\psi(q_g, a_g)}{\psi(q_g, a_g) + \sum_{\substack{q_j \in \mathcal{B}_Q, \\ a_k \in \mathcal{B}_A, \\ \text{ans}(q_j) \neq a_k}} \psi(q_j, a_k)}$$

Joint This method switches from optimizing the probability of single QA pairs to optimizing the *set* of correct QA pairs in the bundle, also known as power-set label classification [108] (Figure 2.3d). We perform this for only bundles consisting of two correct QA pairs, because the power set becomes prohibitively large for larger bundles. Let $C(\mathcal{B})$ be a function that returns all unique subsets of size 2 from the cross product set $\mathcal{B}_Q \times \mathcal{B}_A$, and let (q_{g_1}, a_{g_1}) and (q_{g_2}, a_{g_2}) be the two positive QA pairs in the bundle.

The joint CE objective is

$$\mathcal{L}_{\text{CE-JT}}(\mathcal{B}) = \frac{\psi(q_{g_1}, a_{g_1})\psi(q_{g_2}, a_{g_2})}{\sum_{q_i, a_k, q_j, a_l \in C(\mathcal{B})} \psi(q_i, a_k)\psi(q_j, a_l)}$$

2.2.2 Alternative uses of bundles

Here we briefly consider other potential baselines that make use of instance bundles in some way; we empirically compare against those that are applicable in §2.4.

Data Augmentation If the bundle \mathcal{B} contains instances that were not present in the training data (e.g., the bundle could be generated using simple heuristics; see §2.3), the simplest use of the bundle is to add all instances to the training data and use MLE under the standard IID assumption. This is the standard approach to using this kind of data, and it has been done numerous times previously [109, 110]. This is not applicable if the bundle was obtained by mining the existing training instances, however.

Unlikelihood Unlikelihood training [105] minimizes the likelihood of carefully chosen negative examples to improve a text generation model that would otherwise assign those examples too high of a probability. Essentially, because the generative model only gets a single positive sequence in an exponentially large set, it does not get strong enough evidence to push down the probability of particularly bad generations. Unlikelihood training seeks to solve the same problem that contrastive estimation solves, and it provides a natural alternative use of instance bundles. In our setting, unlikelihood training would decrease the likelihood of negative answers in the bundle:

$$\mathcal{L}_{\text{UL}}(q_g, a_g, \mathcal{B}) = \mathcal{L}_{\text{MLE}}(q_g, a_g) + \sum_{c \in \mathcal{B}_{\mathcal{A}} \setminus a_g} \log(1 - p(c|q_g)) \quad (2.4)$$

The second term in Eq. 2.4 helps provide additional signal to further reduce the likelihood of neighbouring negative answers, especially when the MLE loss starts to overfit at training [111].

Unlikelihood training, though easy to perform, has two drawbacks. First, it independently minimizes the likelihood of the neighborhood, which means that the probability mass is moved from negative QA pairs but may not necessarily move to the positive pair, unlike CE. Second, because it assumes a conditional probabilistic model of $p(a|q)$, it is not clear how to use alternative questions in the bundle with this objective.

2.3 Bundling Heuristics

In this section we discuss how we obtain instance bundles for use with contrastive estimation and other related baselines.

A naive way to create a bundle would be to exploit the fact that all the questions associated with a context are likely to be related, and simply make bundles consisting of all QA pairs associated with the context. However, this approach poses two problems. First, there could be many questions associated with any particular context, and smaller, more closely-related bundles are more informative. Second, and relatedly, it is likely that bundles obtained this way will have many questions whose answers can be obtained from the bundle by superficial type matching. For instance, a wh-question starting with “where” would most likely align with a location type answer. If this were bundled with a question starting with “how many”, with an answer that is a number, the bundle would be largely uninformative. We instead attempt to create bundles with minimally different questions and answers, in several different ways.

Diverse Top-k sampling We first discuss a method for getting alternative *answers* to a single question. This will result in a bundle that can only be used with answer conditional CE, as there are no alternative questions in the bundle. An easy way to get answer candidates is to employ a pre-trained answering model and sample answers from the posterior distribution. However, since the model has seen all the QA pairs while training, it can easily memorize answers, resulting in a low variance, high confidence distribution. In order to achieve diverse answer samples we need to either over-generate and prune out the gold answer from the samples or induce a diversity promoting sampling. We adopt a hybrid sampling strategy where we use nucleus sampling for the first few timesteps (without replacement) and then top-k for the remaining timesteps. This forces the answer generator to consider different starting positions in the passage and then generate the best answer span (of an appropriate length) from the token produced at the first step.

Question Mining Some datasets, such as ROPES, are constructed with very close question pairs already in the data. When these exist, we can create instance bundles by finding natural pairings from the training set. To find these pairings, we cluster the questions with a high lexical bag-of-words overlap based on Jaccard index (≥ 0.8), ensuring that each question in the cluster has a unique answer. In ROPES, these bundles typically result in bundles of two QA pairs that differ in one or a few words. In HotpotQA, the other dataset we focus on in this work, there are very few such pairings naturally occurring in the dataset, so we resort to heuristics to create them.

Question Generation HotpotQA has many questions that are phrased as multiple-choice, with answer options given in the question itself. These multiple choice questions can most often be rephrased to provide QA pairs that can be bundled with the original question. For instance, given the question, “Which animal is faster, turtle or hare?”, it is straightforward to obtain a minimally different question with the opposite answer: “Which animal is slower, turtle or hare?”. We adopt three main heuristics to generate such questions whenever possible, applicable to any dataset that has questions of this kind. All of these heuristics require identifying the two plausible answer choices from the question, which can be done with reasonably high precision using simple regular expressions.

1. We replace superlatives with their contrasting counterparts, e.g., (taller, smaller), (more, less), etc.
2. We negate the main verb, e.g., played \rightarrow didn’t play, by inflecting the verbs.²
3. We swap the noun phrases being compared in the question, e.g., “Are rock A’s wavelengths shorter or longer than rock B’s?” can be used to generate, “Are rock B’s wavelengths shorter or longer than rock A’s?”

²<https://spacy.io/universe/project/lemminflect/>

2.4 Experiments

We use an encoder-decoder style T5-large model for all our experiments. The baseline models in our experiments are the result of fine-tuning the T5 model on the corresponding tasks using the MLE objective. We compare them against models that are further fine-tuned with a combination of MLE and contrastive estimation objectives as described in §2.2.1. That is, when using various instance bundle techniques, we initialize the model with the weights from the fine-tuned MLE model, then continue training with the new loss function.³ The model takes a concatenated context and question as an input to produce an answer output. We use a learning rate of $2e-5$ for ROPES and $5e-5$ for COMPARISON with lowercased inputs and outputs. We truncate the concatenated context and question up to a length of 650 for ROPES and 850 for COMPARISON.

In addition to standard metrics on these datasets, we additionally evaluate using a consistency metric. This metric evaluates to true only if all the questions in a bundle are answered correctly, and is thus a stricter version of EM.

2.4.1 Main results

We experiment with three datasets: a subset of HotpotQA containing only comparison type of questions (COMPARISON), full HotpotQA and ROPES. In general, we find that all variants of CE perform substantially better than MLE alone, with question conditional giving small improvements over other CE variants. All CE models also outperform all UL and data-augmented MLE models.

COMPARISON HotpotQA has several different kinds of questions, with the question category labeled in the original data. We begin by experimenting with the subset labeled as comparison questions, as they lend themselves most naturally to instance bundles. For these questions, we adopt

³To control for the number of optimization steps, we also tried a baseline where we continued fine-tuning an MLE model using the same setup, but this never improved over the original MLE, so we do not include it in the tables.

the question generation strategy to create instance bundles. Table 2.1 shows a comparison of the baseline MLE model (trained on the comparison subset only) with those further fine-tuned with UL and CE over the instance bundles. Also shown is a comparison with further fine-tuning using MLE on the generated QA pairs (+Aug).

	EM	F1	Consistency
MLE (full HotpotQA)	57.4	65.1	36.3
MLE	70.9	77.7	51.2
+ Aug	73.4	80.6	76.7
+ UL	75.1	82.4	85.8
+ Answer Cond.	76.0	83.7	
+ Question Cond.	77.4	84.7	87.3
+ Two way	75.5	82.7	
+ Joint	75.6	83.1	
+ Full Partition	77.4	84.7	

Table 2.1: COMPARISON dev set performance

Due to unavailability of the code for best model on the HotpotQA dataset, we use a T5-large model trained on the entire HotpotQA as a proxy for state of the art model. Even though this model has a performance of 81.1 F1 on the whole dev set (close to the current SOTA 83.5 on the leaderboard⁴), on the comparison subset it performs poorly (65.1). Training an MLE model on just this subset reaches 77.7 F1, which is outperformed by unlikelihood training (82.4 F1). The best CE performance is from the question conditional model, which gets 84.7 F1.

HotpotQA We additionally experiment with the entire HotpotQA dataset. Here we use top-k sampling to create instance bundles, where the top-k answer candidates were sampled from the MLE model we use as a baseline. Table 2.2 shows the performance of the fine-tuned model as we vary the number of answers in $\mathcal{B}_{\mathcal{A}}$ with CE-AC loss. The overall performance gets better with CE up to $|\mathcal{B}_{\mathcal{A}}| = 2$, but reduces after that. On a closer examination of the samples, we find that on average we get two distinct answer candidates and the rest of the candidates are ungrammatical word-piece

⁴<https://hotpotqa.github.io/>

variations of the two distinct candidates (including the oracle answer). These ungrammatical variations provide a noisy signal that hurts model performance.

	F1
MLE	81.1
+CE-AC ($k = 1$)	82.5
+CE-AC ($k = 2$)	83.3
+CE-AC ($k = 3$)	82.1
+CE-AC ($k = 4$)	81.8

Table 2.2: F1 performance on full HotpotQA dev set with increasing number (k) of top- k negative answer candidates

ROPES Since ROPES already contains minimally different QA pairs, we use question mining to create instance bundles. We use as the most closely comparable prior work the multi-step model of [112], which adds a ROPES-specific architecture on top of RoBERTa-large [113].⁵ Our baseline MLE model is a generic T5-large model, with fewer parameters (770M vs. 1.5B) and no special architecture. Table 2.3 shows that using CE gives almost a 12% absolute improvement in EM over an MLE model, and a larger than 12% improvement in consistency, while UL gives only a few point gain.

	EM	Consistency
Multi-step [112]	71.4	-
T5-large MLE	65.7	52.1
+ UL	68.3	55.6
+ Answer Cond.	74.5	
+ Question Cond.	76.6	64.7
+ Two way	73.5	
+ Joint	72.5	
+ Full Partition	75.1	

Table 2.3: ROPES dev set performance

⁵UnifiedQA [74] also evaluated on ROPES, but they used much more training data from many other datasets, and a much larger model than we experiment with, so their performance is not particularly comparable.

2.4.2 Joint Inference

In cases where we can generate a bundle given only a question (that is, the answer candidates are clear and our heuristics can generate a contrasting question), we can treat test time inference as a hard assignment problem between questions and answers in the generated bundle. We use the scoring function $\psi(q, a)$ to align each question to an answer in the bundle by optimizing objective below:

$$\begin{aligned} & \max \sum_{\substack{a_j \in \mathcal{B}_A \\ q_i \in \mathcal{B}_Q}} \psi(q_i, a_j) x_{ij} \\ \text{s.t.} \quad & \sum_{j=0}^{|\mathcal{B}_A|} x_{ij} = 1, \sum_{i=0}^{|\mathcal{B}_Q|} x_{ij} = 1 \end{aligned}$$

We refer to this as joint inference. Intuitively, even if the model is only given a single question at test time, if it can reason jointly about two competing assignments it can potentially use the alternatives to arrive at a better response than if it only considered the single question it was given. As shown in Figure 2.4, when using joint inference the performance of a baseline MLE model on COMPARISON improves from 79 F1 to 85.5. The CE model’s training paradigm manages to achieve this performance (85.8 F1) without enforcing these constraints at test time, but joint inference additionally improves CE, to 90.1 F1.

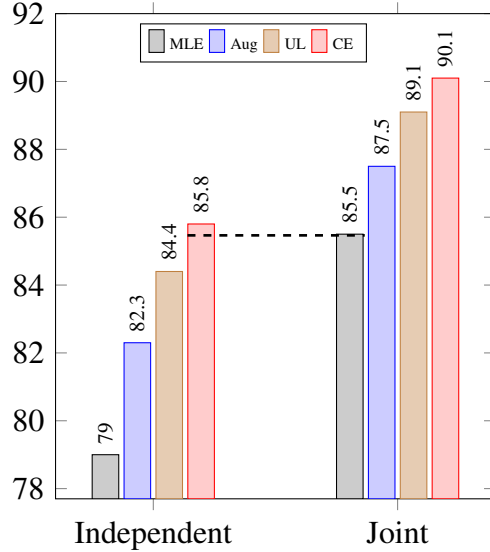


Figure 2.4: Performance (F1) on COMPARISON dev with independent prediction versus joint inference. Joint inference improves all models. The results are on the subset of COMPARISON for which we have paired instances (~93%).

2.4.3 Compatability functions for conditional generation models

We experimented with a few choices of compatibility functions and presented the overall best one in the paper. We describe all the compatibility functions we tried in detail first, and then show results of our experiments on all of them. Our choice of compatibility functions are specific to encoder-decoder style architecture.

A transformer style decoder, d , can be described as a markov random field [114] which takes as input previous answer tokens and question to output current answer token at time step, t , as shown in Figure 2.5. This decoder allows for independent parallel prediction of tokens at each time step, which make the answer likelihood a product of independent markov random fields, $\prod_t p(a_t|a_{<t}, q)$, tied by same parameters in function d .

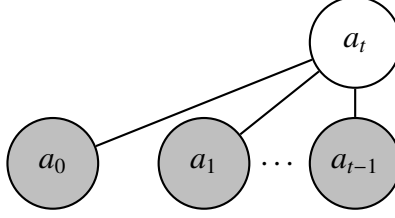


Figure 2.5: Answer decoder

Locally-normalized (LN) scores can be defined as sum of log-probability of token at each time step locally normalized over vocabulary.

$$\begin{aligned}
 \psi_L(q, a) &= \exp \log \prod_t p(a_t | a_{<t}, q) \\
 &= \exp \sum_t \log \frac{\exp(d(a_t; a_{<t}, q))}{\sum_{v \in |V|} \exp(d(v; a_{<t}, q))}
 \end{aligned} \tag{2.5}$$

Un-normalized scores (UN): Locally normalized scores may suffer from label bias [115] which can be crucial when contrastive answers that have overlapping subsequences, for eg., if answer candidates are a list of choices {"Person A", "Person B", "Person C"}.

$$\begin{aligned}
 \psi_U(q, a) &= \exp \log \prod_t \exp(d(a_t; a_{<t}, q)) \\
 &= \exp \log \exp\left(\sum_t d(a_t; a_{<t}, q)\right) \\
 &= \exp \sum_t d(a_t; a_{<t}, q)
 \end{aligned} \tag{2.6}$$

Whole Sequence (GS): score considers the score of the last token, which is often a special symbol like, $\langle eos \rangle$ tag. Intuitively, this can be seen as a score for the entire input answer sequence,

$$\psi_G(q, a) = d(a_T; a_{<T}, q)$$

2.4.4 Ablation results

From Table 2.4 and Table 2.5 we can see that the question conditional loss overall works well. Interestingly GS compatibility function performs well on COMPARISON but not on ROPES. We conjecture that’s because in ROPES often the candidates have very small lexical dissimilarities (e.g., “Patient A”, “Patient B”) which makes looking at each token important. In ROPES, full partition loss performs slightly better than question conditions.

	Single Neighborhood		Multiple Neighborhood		
	Answer Conditional	Question Conditional	Two-way	Joint	Full Partition
LN	76.0/83.7	77.4/84.7	75.5/82.7	75.6/83.1	73.2/81.1
UN	75.7/82.9	76.6/84.6	75.3/82.6	75.6/83.3	76.7/84.7
GS	76.9/84.1	76.4/83.6	76.0/83.2	74.7/81.5	75.7/83.3

Table 2.4: Performance (EM/F1) of COMPARISON dev set on models finetuned with different single and multiple neighborhood CE losses.

	Single Neighborhood		Multiple Neighborhood		
	Answer Conditional	Question Conditional	Two-way	Joint	Full Partition
LN	74.5	76.6	73.5	72.5	75.1
UN	73.0	75.9	74.9	71.2	77.6
GS	71.8	69.5	71.1	69.7	73.1

Table 2.5: Performance (EM) of ROPES dev set on models finetuned with different single and multiple neighborhood CE losses. We do not report F1 as it’s not a good performance measure on this dataset

2.5 Discussion

In this section we try to understand how CE compares to MLE and UL and under what conditions it is effective.

2.5.1 Relation between MLE, UL and CE

In this section we try to understand the relation between CE, MLE and UL in the special case when the scoring function ψ comes from a locally-normalized generative model, as it does in this work. Let p_V be the locally normalized probability of an answer candidate (i.e., the combined likelihood of each token in a sequence under a given generative model). $\psi(q, a)$ then equals $p_V(a|q)$. The CE-AC loss with locally normalized compatibility score can be described as

$$\mathcal{L}_{\text{CE-AC}}(q_g, a_g, \mathcal{B}) = \log \frac{p_V(a_g|q_g)}{\sum_{c \in \mathcal{B}_{\mathcal{A}}} p_V(a_c|q_g)} \quad (2.7)$$

We can decompose and rewrite Eq. 2.7 as

$$\begin{aligned} \mathcal{L}_{\text{CE-AC}}(q_g, a_g, \mathcal{B}) &= \log p_V(a_g|q_g) - \log \sum_{c \in \mathcal{B}_{\mathcal{A}}} p_V(a_c|q_g) \\ &= \mathcal{L}_{\text{MLE}}(q_g, a_g) + \text{Reg}(\mathcal{B}_{\mathcal{A}}, q_g) \end{aligned} \quad (2.8)$$

Eq. 2.8 shows that $\mathcal{L}_{\text{CE-AC}}$ is just a linear combination of MLE and a regularization term which decreases the probability of each incorrect answer in the bundle.

On a closer look we can see an interesting connection between the regularization term and unlikelihood. The regularization term in CE-AC is essentially the log of an unlikelihood term, except the unlikelihood objective in §2.2.2 in practice gets applied at each timestep of decoding, while the regularization term in CE-AC is applied over the entire answer sequence.

Our formulation of CE is more general than the specific case we are analyzing here, but we make note of it as this is the function that we used in our experiments, and it significantly outperformed unlikelihood training. The theoretical connections shown here could benefit from further exploration.

2.5.2 The importance of close instance bundles

Experiments on ROPES and COMPARISON show strong improvements by using CE and UL when instances can be grouped into very closely related bundles. But such effective grouping may not be possible on all datasets. To analyze the applicability of our methods to a dataset without natural bundles, we looked at QUOREF [116]. Table 2.6 shows a comparison between the trends of improvements due to UL and CE across QUOREF, ROPES and COMPARISON with bundles created using top-k sampling. As it can be seen from the results, UL does not improve on top of MLE, and CE shows only a very small improvement which is likely statistical noise. To understand why, we analyzed the $p(a|q, c)$ distribution of the baseline MLE model, and computed the following two measures on a random sample of the training set.

- Entropy₁₀ = $-\sum_{i=1}^{10} p(a_i|q, c) \log p(a_i|q, c)$
- Top-2 ratio = $\log p(a_1|q, c)/p(a_2|q, c)$

As seen in Table 2.6, QUOREF has a lower Entropy₁₀, and a higher Top-2 ratio than the other datasets, indicating that the baseline MLE model places a lot more weight on the top-1 answer in this task. Manual analysis additionally found that most of the top predictions were ungrammatical variations of the top-1 answer, similar to (but more extreme than) what was seen on the full HotpotQA dataset. This could explain why the top-k bundling heuristic is not as effective in the case of QUOREF as the other two datasets. More generally, these results indicate the importance of effective instance bundling heuristics, and future work could focus on identifying more general ways to create bundles.

2.6 Related Work

Learning with negative samples has been explored in many natural language tasks, such as dialogue generation [117], word embeddings [118], language modeling [119], etc., and computer vision

Dataset	MLE	UL	CE	Entropy ₁₀	Top-2 ratio
COMP.	77.7	82.4	84.7	2.31	0.5
ROPES	65.7	68.3	77.6	0.97	2.3
QUOREF	84.8	83.9	85.0	0.06	3.1

Table 2.6: Comparison between QUOREF, COMPARISON, and ROPES datasets with Top-k bundling. MLE, UL, and CE results are on the corresponding development sets (F_1 for COMPARISON and QUOREF, EM for ROPES) and Entropy₁₀ and Top-2 ratio are measured on random samples of the training sets. UL and CE columns show results after fine-tuning the baseline MLE model with the respective objectives.

tasks such as image captioning [120], unsupervised representation learning [121], etc. In similar vein, mutual information minimization based learners in question answering [122] and image classification [123] try to decrease the mutual information between positive and negative samples.

Natural language applications often sample negative examples either randomly from the data or based on likelihood (or unlikelihood) metrics from a reference model. However, the negative samples extracted in this manner are often unrelated. A growing body of literature is exploring ways to obtain closely-related examples, either manually [124, 11] or automatically [125, 126, 127]. This trend is complementary to our work, as we show how to make better use of these closely-related examples during training. There is also work on consistent cluster assignments in co-reference resolution [128]; factually consistent summaries [129] and language models [130].

There is also a growing body of literature on training with closely related examples, to which we are contributing. Several works make use of logical consistency in natural language inference tasks [131, 132, 102]. Another line of work [133, 134, 135, 136] tries to increase (or decrease) the distance between intermediate representations of contrasting (or paraphrased) instances.

Chapter 3

Generative Question Answering

By their very nature, heuristic shortcuts will produce biases, and that is true for both humans and artificial intelligence, but the heuristics of AI are not necessarily the human ones.

Daniel Kahneman

(Nobel Memorial Prize Awardee)

Compositional reasoning tasks like multi-hop question answering, require making latent decisions to get the final answer, given a question. However, crowdsourced datasets often capture only a slice of the underlying task distribution, which can induce unanticipated biases in models performing compositional reasoning. Furthermore, discriminatively trained models exploit such biases to get a better held-out performance, without learning the right way to reason, as they do not necessitate paying attention to the question representation (conditioning variable) in its entirety, to estimate the answer likelihood. In this work, we propose a generative context selection model for multi-hop question answering that reasons about how the given question could have been generated

given a context pair. While being comparable to the state-of-the-art answering performance, our proposed generative passage selection model has a better performance (4.9% higher than baseline) on adversarial held-out set which tests robustness of model’s multi-hop reasoning capabilities.

3.1 Introduction

Recently many reading comprehension datasets like HotpotQA [40] and WikiHop [137] that require compositional reasoning over several disjoint passages have been introduced. This style of compositional reasoning, also referred to as multi-hop reasoning, first requires finding the correct set of passages relevant to the question and then the answer span in the selected set of passages. Most of these dataset are often collected via crowdsourcing, which makes the evaluation of such models heavily reliant on the quality of the collected held-out sets.

Crowdsourced datasets often present only a partial picture of the underlying data distribution. Learning complex latent sequential decisions, like multi-hop reasoning, to answer a given question under such circumstances is marred by numerous biases, such as annotator bias [138], label bias [13, 139], survivorship bias [140, 2], and ascertainment bias [89]. As a result, testing model performance on such biased held-out sets becomes unreliable as the models exploit these biases and learn shortcuts to get the right answer but without learning the right way to reason.

Consider an example from HotpotQA in Figure 3.1, where the latent entity “Virginia Commonwealth University” can be used by the model [2] to bridge the two relevant passages (highlighted in green) from the original dev set and correctly predict the answer “1838”. However, upon adding an adversarial context (highlighted in pink) to the pool of contexts, the model prediction changes to “1938” implying that the model did not learn the right way to reason. This is because the discriminatively trained passage selector exploits lexical cues like “founded” in the second passage and does not pay attention to the complete question. The absence of such adversarial contexts at

Question: The 2011-12 VCU Rams men’s basketball team, led by third year head coach Shaka Smart, represented the university which was founded in what year?

Gold Answer: 1838

Passage 1: The 2011-12 VCU Rams men’s basketball team represented Virginia Commonwealth University during the 2011-12 NCAA Division I men’s basketball season...

Passage 2: Virginia Commonwealth University (VCU) is a public research university located in Richmond, Virginia. VCU was founded in 1838 as the medical department of Hampden-Sydney College, becoming the Medical College of Virginia in 1854...

Prediction: 1838

Adversarial context from [2]:

Dartmouth University is a public research university located in Richmond, Virginia. Dartmouth was founded in 1938 as the medical department of Hampden-Sydney College, becoming the Medical College of Virginia in 1854...

New Prediction: 1938

Figure 3.1: Example from HotpotQA, showing the reasoning chain for answering the question (in green) and an adversarial context (in pink) introduced by [2] which confuses the model, causing it to change its prediction because it did not learn the right way to reason.

training allows the model to find incorrect reasoning paths.

In this work, we propose a generative context pair selection model, which tries to reason through the data generation process of how a specific question could have been constructed from a given pair of passages. We show that our proposed model is comparable in performance to the state-of-the-art systems, with minimal drop in performance on the adversarial held-out set. Our generative passage selector shows an improvement of 4.9% in Top-1 accuracy as compared to discriminatively trained passage selector on the adversarial dev set.

3.2 Generative Passage Selection

Given a set of contexts $C = \{c_0, c_1, \dots, c_N\}$, the goal of multi-hop question answering is to combine information from multiple context passages to identify the answer span a for a given question q . In

single-hop QA, the goal is to identify the *pair* of contexts, from all possible pairs $\psi = \{(c_i, c_j) : c_i \in C, c_j \in C\}$, that is appropriate for answering the question.

Existing models for multi-hop question answering [3, 141] consist of two components: a *discriminative passage selection* and an *answering model*. Passage selection identifies which contexts are relevant for answering the given question, i.e. estimates $p(c_{ij}|q, \psi)$. This is followed by the answering model to extract the answer span given a context pair and the question ($p(a|q, c_{ij})$). These are combined as follows:

$$p(a|q, \psi) = \sum_{c_{ij}} p(a|q, c_{ij})p(c_{ij}|q, \psi) \tag{3.1}$$

The discriminative passage selector learns to select a set of contexts by conditioning on the question representation. This learning process does not encourage the model to pay attention to the entire question, which can result in ignoring parts of the question, thus, learning spurious correlations.

To predict the answer at test time, we do not sum over all pairs of contexts, but instead use the top scoring pair to answer the question¹.

In other words, we use *passage selection* to pick the best context pair c_{ij}^* , which is used by the answering module to get the answer, $a^* = \operatorname{argmax} p(a|q, c_{ij}^*)$.

3.2.1 Model Description

We propose a joint question-answering model which learns $p(a, q|\psi)$ instead of $p(a|q, \psi)$. This joint question-answer model can be factorized into a generative passage selector and a standard

¹Summing over all context pairs, or maintaining a beam of highly ranked pairs, did not yield much higher performance, in particular, not worth the additional computation cost.

answering model as:

$$p(a, q|\psi) = \sum_{c_{ij}} p(a|q, c_{ij})p(q|c_{ij})p(c_{ij}|\psi) \quad (3.2)$$

First, a prior, $p(c_{ij}|\psi)$, over the context pairs establishes a measure of compatibility between passages in a particular dataset. Then, a conditional generation model, $p(q|c_{ij})$, establishes the likelihood of generating the given question from a selected pair of passages. Finally, a standard answering model, $p(a|q, c_{ij})$, estimates the likely answer distribution given a question and context pair. The first two terms (prior and conditional generation) can be seen as a generative model that chooses a pair of passages from which the given question could have been constructed. The answering model can be instantiated with any existing state-of-the-art model, such as a graph neural network [3, 142], entity-based chain reasoning [141], etc.

The process at test time is identical to that with discriminative passage selection, except that the context pairs are scored by taking the entire question into account, $c_{ij}^* = \operatorname{argmax}_{c_{ij}} p(q|c_{ij})p(c_{ij}|\psi)$.

3.2.2 Model Learning

We use a pre-trained T5 [58] based encoder-decoder model for obtaining contextual representations, which are further trained to estimate all individual probability distributions.

For learning the generative model, we train the prior, $p(c_{ij}|\psi)$ and the conditional generation model $p(q|c_{ij}, \psi)$ jointly. First, the prior network projects the concatenated contextualized representation, r_{ij} , of starting and ending token of concatenated contexts $(c_i; c_j)$, from the encoder to obtain unnormalized scores, which are then normalized across all context-pairs via softmax operator. The

Dataset	Standard Selector	Generative Selector
	$p(c_{ij} q, \psi)$	$p(q c_{ij})p(c_{ij} \psi)$
HotpotQA	95.3	97.5
WikiHop	96.8	97.2

Table 3.1: **Passage selection accuracy:** Accuracy that the selected passage pair (c_{ij}^*) by different techniques is the oracle one (c_{gold}) on original development set.

loss function tries to increase the likelihood of gold context pair over all possible context pairs.

$$r_{ij} = \text{encoder}(c_i; c_j) \quad (3.3)$$

$$s_{ij} = W^{1 \times d}(r_{ij}[\text{start}]; r_{ij}[\text{end}]) \quad (3.4)$$

The conditional question generation network gets contextual representations for context-pair candidates from the encoder and uses them to generate the question, via the decoder. We define the objective to increase the likelihood of the question for gold context pairs and the unlikelihood [105] for a sample set of *negative* context pairs (Eq. 3.5)

$$\begin{aligned} \mathcal{L}(\theta) = & \sum_{t=1}^{|question|} \log p(q_t|q_{<t}, c_{gold}) \\ & + \sum_{n \in |neg.pairs|} \sum_{t=1}^{|question|} \log(1 - p(q_t|q_{<t}, c_n)) \end{aligned} \quad (3.5)$$

3.3 Experiments and Results

We experiment with two popular multi-hop datasets: HotpotQA [40] and WikiHop [137]. Most SOTA passage selection modules for HotpotQA use a RoBERTa [113] based classifier to select top-k passages given the question, which has an accuracy of $\sim 94.5\%$ [3]. We used a T5-based standard passage selector, $p(c_{ij}|q, \psi)$, as our baseline, which provides a comparable performance to SOTA passage selector (Table 3.1).

Models	Original		Adversarial	
	Acc	F1	Acc	F1
Standard Selector	95.3	79.5	91.4	76.0
Generative Selector	97.5	81.9	96.3	80.1
[3]	94.5	80.2	-	61.1
[4]	-	82.2	-	78.9

Table 3.2: **Performance on Adversarial Data:** Passage selection accuracy and end to end QA F1 on original and adversarial set [2] of HotpotQA. The results of [3] and [4] are taken from [5].

We also use a simple T5-based answering model that has a comparable performance to SOTA answering models to illustrate the effect of our generative selector on end-to-end model performance. The *oracle* EM/F1 of our answering model, $p(a|q, c_{gold})$, on HotpotQA and WikiHop are 74.5/83.5 and 76.2/83.9 respectively. The overall EM/F1 of WikiHop with generative model are 73.5/80.2.

3.3.1 Adversarial Evaluation

We use an existing adversarial set [2] for HotpotQA to test the robustness of model’s multi-hop reasoning capabilities given a confusing passage. This helps measure, quantitatively, the degree of biased correlations learned by the model. In Table 3.2, we show that the standard discriminative passage selector has a much higher performance drop ($\sim 4\%$) as compared to the generative selector ($\sim 1\%$) on adversarial dev set [2], showing that generative selector is less biased and less affected by conservative changes [143] to the data distribution. We can also see in Table 3.2 that SOTA models [3, 4], which use the standard passage selector, also have a larger F1 drop when applied to the adversarial set. Table 3.3 shows that the generator was able to generate multi-hop style questions using both the contexts.

Context 1, c_i:	The America East Conference is a collegiate athletic conference affiliated with the NCAA Division I, whose members are located mainly in the Northeastern United States. The conference was known as the Eastern College Athletic Conference-North from 1979 to 1988 and the North Atlantic Conference from 1988 to 1996.
Context 2, c_j:	The Vermont Catamounts men’s soccer team represents the University of Vermont in all NCAA Division I men’s college soccer competitions. The team competes in the America East Conference.
Original Question, q:	the vermont catamounts men’s soccer team currently competes in a conference that was formerly known as what from 1988 to 1996?
Generated Questions: $p(q c_{ij}, \psi)$	<p>the vermont catamounts men’s soccer team competes in what collegiate athletic conference affiliated with the ncaa division i, whose members are located mainly in the northeastern united states?</p> <p>the vermont catamounts men’s soccer team competes in a conference that was known as what from 1979 to 1988?</p> <p>the vermont catamounts men’s soccer team competes in a conference that was known as what from 1988 to 1996?</p>

Table 3.3: Sample questions generated by using the question generation decoder with top-k sampling show that the generative model is able to construct (reason about) possible multi-hop questions given a context-pair.

3.3.2 Context pairs vs. Sentences

Some context selection models for HotpotQA use a multi-label classifier that chooses top-k sentences [4, 144] which result in limited inter-document interaction than context pairs. To compare these two input types, we construct a multi-label sentence classifier $p(s|q, C)$ that selects relevant sentences. This classifier projects a concatenated sentence and question representation, followed by a sigmoid, to predict if the sentence should be selected.

This model has a better performance over the context-pair selector but is more biased (Table 3.4).

We performed similar experiments with the generative model. Along with the *passage* selection model, we train a generative *sentence* selection model by first selecting a set of sentences with gumbel softmax and then generating the question given the set of sentences.

Given that the space of set of sentences is much larger than context pairs, the generative sentence selector does not have good performance (Table 3.4).

Model	Original	Adversarial
Discriminative Selectors		
Passage, $p(c_{ij} q, \psi)$	95.3	96.3
Sentence, $p(s q, C)$	97.6	90.9
Generative Selectors		
Passage, $p(q c_{ij}, \psi)p(c_{ij} \psi)$	97.5	96.3
Sentence, $p(q s, C)p(s C)$	90.6	89.2
Multi-task, $p(q, s c_{ij}, \psi)p(c_{ij} \psi)$	98.1	97.2

Table 3.4: **Passages vs Sentences:** Passage selection accuracy for models with different context inputs on the development and adversarial set of HotpotQA.

To further improve the performance of the generative selector, we add an auxiliary loss term that predicts the relevant sentences in the context pair, $p(q, s|c_{ij}, \psi)$, along with selecting the context pair in a multi-task setting. We see slight performance improvements by using relevant sentences as an additional supervision signal.

3.4 Conclusion

We have presented a generative formulation of context pair selection in multi-hop question answering models. By encouraging the context selection model to *explain* the entire question, it is less susceptible to bias, performing substantially better on adversarial data than existing methods that use discriminative selection. Our proposed model is simple to implement and can be used with *any* existing (or future) answering model; we will release code to support this integration.

Since context pair selection scales quadratically with the number of contexts, it is not ideal for scenarios that involve a large number of possible contexts. However, it allows for deeper inter-document interaction as compared to other approaches that use summarized document representations. With more reasoning steps, selecting relevant documents given only the question becomes challenging, increasing the need for inter-document interaction.

3.5 Related work

Most passage selection models for HotpotQA and Wikihop’s distractor style setup employ a RoBERTA based context selectors given the question [3, 4]. In an ideal scenario, the absence of latent entity in the question should not allow selection of all oracle passages. However, the high performance of these systems can be attributed to existing bias in HotpotQA [2, 140]. Another line of work dynamically updates the working memory to re-rank the set of passage at each hop [145].

With the release of datasets like SearchQA [146], TriviaQA [147], and NaturalQuestions [148], lot of work has been done in open-domain passage retrieval, especially in the full Wikipedia setting. However, these questions do not necessarily require multi-hop reasoning. A series of work has tried to match a document-level summarized embedding to the question [149, 150, 151] for obtaining the relevant answers. In generative question answering, a few works [111, 152] have used a joint question answering approach on single context.

Part II

Decomposition: Understanding Compositionality

Chapter 4

DROP: Discrete Reasoning Over Passages

It is a capital mistake to theorize before one has data.

Sir Arthur Conan Doyle

(British writer and physician)

In order to evaluate the compositional ability of neural models via a controlled test, we introduce a new English reading comprehension benchmark dataset that necessitates a series (composition) of operations to solve the question. This dataset, which we refer to as DROP, requires performing **Discrete Reasoning Over** the content of **Paragraphs**. Within this benchmark, which comprises 96K questions created through crowdsourcing and adversarial techniques, a system is required to handle references within a question, potentially to multiple input positions, and execute discrete compositional operations on them. These operations may include addition, counting, or sorting. These operations require a much more comprehensive understanding of the content of paragraphs than what was necessary for prior datasets. We apply state-of-the-art methods from both the reading comprehension and semantic parsing literatures on this dataset and show that the best systems only achieve 32.7% F_1 on our generalized accuracy metric, while expert human performance is 96.4%.

We additionally present a new model that combines reading comprehension methods with simple numerical reasoning to achieve 47.0% F_1 .

4.1 Introduction

Reading comprehension has recently seen rapid progress, with systems matching humans on the most popular datasets for the task. However, a large body of work has highlighted the brittleness of these systems, showing that there is much work left to be done. We introduce a new English reading comprehension benchmark, DROP, which requires **D**iscrete **R**easoning **O**ver the content of **P**aragraphs. In this crowdsourced, adversarially-created, 96k-question benchmark, a system must resolve references in a question, perhaps to multiple input positions, and perform discrete operations over them (such as addition, counting, or sorting). These operations require a much more comprehensive understanding of the content of paragraphs than what was necessary for prior datasets. We apply state-of-the-art methods from both the reading comprehension and semantic parsing literatures on this dataset and show that the best systems only achieve 32.7% F_1 on our generalized accuracy metric, while expert human performance is 96.4%. We additionally present a new model that combines reading comprehension methods with simple numerical reasoning to achieve 47.0% F_1 .

The task of *reading comprehension*, where systems must understand a single passage of text well enough to answer arbitrary questions about it, has seen significant progress in the last few years, so much that the most popular datasets available for this task have been solved [153, 32]. We introduce a substantially more challenging English reading comprehension dataset aimed at pushing the field towards more comprehensive analysis of paragraphs of text. In this new benchmark, which we call DROP, a system is given a paragraph and a question and must perform some kind of Discrete Reasoning Over the text in the Paragraph to obtain the correct answer.

These questions that require discrete reasoning (such as addition, sorting, or counting; see Table 4.1) are inspired by the complex, compositional questions commonly found in the semantic parsing literature. We focus on this type of questions because they force a structured analysis of the content of the paragraph that is detailed enough to permit reasoning. Our goal is to further *paragraph understanding*; complex questions allow us to test a system’s understanding of the paragraph’s semantics.

DROP is also designed to further research on methods that combine distributed representations with symbolic, discrete reasoning. In order to do well on this dataset, a system must be able to find multiple occurrences of an event described in a question (presumably using some kind of soft matching), extract arguments from the events, then perform a numerical operation such as a sort, to answer a question like “*Who threw the longest touchdown pass?*”.

We constructed this dataset through crowdsourcing, first collecting passages from Wikipedia that are easy to ask hard questions about, then encouraging crowd workers to produce challenging questions. This encouragement was partially through instructions given to workers, and partially through the use of an *adversarial baseline*: we ran a baseline reading comprehension method (BiDAF) [49] in the background as crowd workers were writing questions, requiring them to give questions that the baseline system could not correctly answer. This resulted in a dataset of 96,567 questions from a variety of categories in Wikipedia, with a particular emphasis on sports game summaries and history passages. The answers to the questions are required to be spans in the passage or question, numbers, or dates, which allows for easy and accurate evaluation metrics.

We present an analysis of the resulting dataset to show what phenomena are present. We find that many questions combine complex question semantics with SQuAD-style argument finding; e.g., in the first question in Table 4.1, BiDAF correctly finds the amount the painting sold for, but does not understand the question semantics and cannot perform the numerical reasoning required to answer the question. Other questions, such as the fifth question in Table 4.1, require finding all events in the passage that match a description in the question, then aggregating them somehow (in

this instance, by counting them and then performing an argmax). Very often entity coreference is required. Table 4.1 gives a number of different phenomena, with their proportions in the dataset.

We used three types of systems to judge baseline performance on DROP: (1) heuristic baselines, to check for biases in the data; (2) SQuAD-style reading comprehension methods; and (3) semantic parsers operating on a pipelined analysis of the passage. The reading comprehension methods perform the best, with our best baseline achieving 32.7% F_1 on our generalized accuracy metric, while expert human performance is 96.4%. Finally, we contribute a new model for this task that combines limited numerical reasoning with standard reading comprehension methods, allowing the model to answer questions involving counting, addition and subtraction. This model reaches 47% F_1 , a 14.3% absolute increase over the best baseline system.

4.2 Data Collection

In this section, we describe our annotation protocol, which consists of three phases. First, we automatically extract passages from Wikipedia which are expected to be amenable to complex questions. Second, we crowdsource question-answer pairs on these passages, eliciting questions which require discrete reasoning. Finally, we validate the development and test portions of DROP to ensure their quality and report inter-annotator agreement.

Passage extraction We searched Wikipedia for passages that had a narrative sequence of events, particularly with a high proportion of numbers, as our initial pilots indicated that these passages were the easiest to ask complex questions about. We found that National Football League (NFL) game summaries and history articles were particularly promising, and we additionally sampled from any Wikipedia passage that contained at least twenty numbers.¹ This process yielded a collection of about 7,000 passages.

¹We used an October 2018 Wikipedia dump, as well as scraping of online Wikipedia.

Question collection We used Amazon Mechanical Turk² to crowdsource the collection of question-answer pairs, where each question could be answered in the context of a single Wikipedia passage. In order to allow some flexibility during the annotation process, in each human intelligence task (HIT) workers were presented with a random sample of 5 of our Wikipedia passages, and were asked to produce a total of at least 12 question-answer pairs on any of these.

We presented workers with example questions from five main categories, inspired by questions from the semantic parsing literature (addition/subtraction, minimum/maximum, counting, selection and comparison; see examples in Table 4.1), to elicit questions that require complex linguistic understanding and discrete reasoning. In addition, to further increase the difficulty of the questions in DROP, we employed a novel adversarial annotation setting, where workers were only allowed to submit questions which a real-time QA model BiDAF *could not* solve.³

Next, each worker answered their own question with one of three answer types: spans of text from either question or passage, a date (which was common in history and open-domain text) and numbers, allowed only for questions which explicitly stated a specific unit of measurement (e.g., “How many *yards* did Brady run?”), in an attempt to simplify the evaluation process.

Initially, we opened our HITs to all United States workers and gradually reduced our worker pool to workers who understood the task and annotated it well. Each HIT paid 5 USD and could be completed within 30 minutes, compensating a trained worker with an average pay of 10 USD/ hour.

Overall, we collected a total of 96,567 question-answer pairs with a total Mechanical Turk budget of 60k USD (including validation). The dataset was randomly partitioned by passage into training (80%), development (10%) and test (10%) sets, so all questions about a particular passage belong to only one of the splits.

²www.mturk.com

³While BiDAF is no longer state-of-the-art, performance is reasonable and the AllenNLP implementation [154] made it the easiest to deploy as a server.

Validation In order to test inter-annotator agreement and to improve the quality of evaluation against DROP, we collected at least two additional answers for each question in the development and test sets.

In a separate HIT, workers were given context passages and a previously crowdsourced question, and were asked to either answer the question or mark it as invalid (this occurred for 0.7% of the data, which we subsequently filtered out).

We found that the resulting inter-annotator agreement was good and on par with other QA tasks; overall Cohen’s κ was 0.74, with 0.81 for numbers, 0.62 for spans, and 0.65 for dates.

4.3 Data Analysis

In the following, we quantitatively analyze properties of passages, questions, and answers in DROP. Different statistics of the dataset are depicted in Table 4.2. Notably, questions have a diverse vocabulary of around 30k different words in our training set.

Question analysis To assess the question type distribution, we sampled 350 questions from the training and development sets and manually annotated the categories of discrete operations required to answer the question. Table 4.1 shows the distribution of these categories in the dataset. In addition, to get a better sense of the lexical diversity of questions in the dataset, we find the most frequent trigram patterns in the questions per answer type. We find that the dataset offers a huge variety of linguistic constructs, with the most frequent pattern (“Which team scored”) appearing in only 4% of the span type questions. For number type questions, the 5 most frequent question patterns all start with “How many”, indicating the need to perform counting and other arithmetic operations. A distribution of the trigrams containing the start of the questions are shown in Figure 4.1.

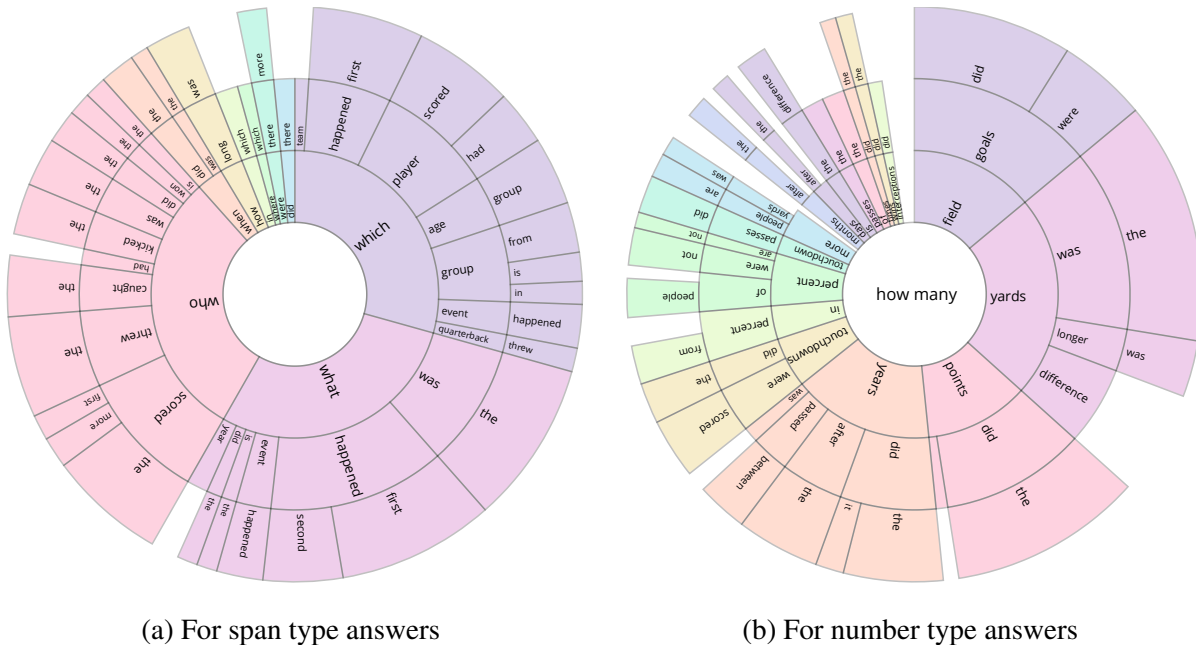


Figure 4.1: Distribution of the most popular question prefixes for two different subsets of the training data.

Answer analysis To discern the level of *passage understanding* needed to answer the questions in DROP, we annotate the set of spans in the passage that are necessary for answering the 350 questions mentioned above. We find that on an average 2.18 spans need to be considered to answer a question and the average distance between these spans is 26 words, with 20% of samples needing at least 3 spans (see appendix for examples). Finally, we assess the answer distribution in Table 4.3, by running the part-of-speech tagger and named entity recognizer from spaCy⁴ to automatically partition all the answers into various categories. We find that a majority of the answers are numerical values and proper nouns.

4.3.1 Baseline Systems

In this section we describe the initial baselines that we evaluated on the DROP dataset. We used three types of baselines: state-of-the-art semantic parsers (§4.3.2), state-of-the-art reading

⁴<https://spacy.io/>

comprehension models (§4.3.3), and heuristics looking for annotation artifacts (§4.3.4). We use two evaluation metrics to compare model performance: *Exact-Match*, and a numeracy-focused (macro-averaged) F_1 score, which measures overlap between a bag-of-words representation of the gold and predicted answers. We employ the same implementation of Exact-Match accuracy as used by SQuAD, which removes articles and does other simple normalization, and our F_1 score is based on that used by SQuAD. Since DROP is numeracy-focused, we define F_1 to be 0 when there is a number mismatch between the gold and predicted answers, regardless of other word overlap. When an answer has multiple spans, we first perform a one-to-one alignment greedily based on bag-of-word overlap on the set of spans and then compute average F_1 over each span. When there are multiple annotated answers, both metrics take a max over all gold answers.

4.3.2 Semantic Parsing

Semantic parsing has been used to translate natural language utterances into formal executable languages (e.g., SQL) that can perform discrete operations against a structured knowledge representation, such as knowledge graphs or tabular databases [155, 156, 157, 158].

Since many of DROP’s questions require similar discrete reasoning, it is appealing to port some of the successful work in semantic parsing to the DROP dataset. Specifically, we use the grammar-constrained semantic parsing model built by [159] (KDG) for the WIKITABLEQUESTIONS tabular dataset [160].

Sentence representation schemes We experimented with three paradigms to represent paragraphs as structured contexts:

(1) Stanford dependencies [161]; which capture word-level syntactic relations, (2) Open Information Extraction [94], a shallow semantic representation which directly links predicates and arguments;

and (3) Semantic Role Labeling [162], which disambiguates senses for polysemous predicates and assigns predicate-specific argument roles.⁵ To adhere to KDG’s structured representation format, we convert each of these representations into a table, where rows are predicate-argument structures and columns correspond to different argument roles.

Logical form language Our logical form language identifies five basic elements in the table representation: *predicate-argument structures* (i.e., table rows), *relations* (column-headers), *strings*, *numbers*, and *dates*. In addition, it defines functions that operate on these elements, such as counters and filters.⁶

Following [159], we use the argument and return types of these functions to automatically induce a grammar to constrain the parser.

We also add context-specific rules to produce strings occurring in both question and paragraph, and those paragraph strings that are neighbors of question tokens in the GloVe embedding space [166], up to a cosine distance of d .⁷ The complete set of functions used in our language and their induced grammar can be found in the code release.

Training and inference During training, the KDG parser maximizes the marginal likelihood of a set of (possibly spurious) question logical forms that evaluate to the correct answer. We obtain this set by performing an exhaustive search over the grammar up to a preset tree depth. At test time, we use beam search to produce the most likely logical form, which is then executed to predict an answer.

⁵We used the AllenNLP implementations of state-of-the-art models for all of these representations [154, 163, 164, 165].

⁶For example `filter_number_greater` takes a set of predicate-argument structures, the name of a relation, and a number, and returns all those structures where the numbers in the argument specified by the relation are greater than the given number.

⁷ $d = 0.3$ was manually tuned on the development set.

4.3.3 SQuAD-style Reading Comprehension

We test four different SQuAD-style reading comprehension models on DROP: (1) **BiDAF** [49], which is the adversarial baseline we used in data construction (66.8% EM on SQuAD 1.1); (2) **QANet** [50], currently the best-performing published model on SQuAD 1.1 without data augmentation or pre-training (72.7% EM); (3) **QANet + ELMo**, which enhances the QANet model by concatenating pre-trained ELMo representations [167] to the original embeddings (78.7% EM); (4) **BERT** [32], which recently achieved improvements on many NLP tasks with a novel pre-training technique (84.7% EM).⁸

These models require a few minor adaptations when training on DROP. While SQuAD provides answer indices in the passage, our dataset only provides the answer strings. To address this, we use the marginal likelihood objective function proposed by [168], which sums over the probabilities of all the matching spans.⁹ We also omitted the training questions which cannot be answered by a span in the passage (45%), and therefore cannot be represented by these systems.

For the BiDAF baseline, we use the implementation in AllenNLP but change it to use the marginal objective. For the QANet model, our settings differ from the original paper only in the batch size (16 v.s. 32) and number of blocks in the modeling layer (6 v.s. 7) due to the GPU memory limit. We adopt the ELMo representations trained on 5.5B corpus for the QANet+ELMo baseline and the large uncased BERT model for the BERT baseline. The hyper-parameters for our NAQANet model (§4.4) are the same as for the QANet baseline.

⁸The first three scores are based on our own implementation, while the score for BERT is based on an open-source implementation from Hugging Face: <https://github.com/huggingface/pytorch-pretrained-bert>

⁹For the black-box BERT model, we convert DROP to SQuAD format by using the first match as the gold span.

4.3.4 Heuristic Baselines

A recent line of work [91, 169] has identified that popular crowdsourced NLP datasets (such as SQuAD [170] or SNLI [171]) are prone to have artifacts and annotation biases which can be exploited by supervised algorithms that learn to pick up these artifacts as signal instead of more meaningful semantic features.

We estimate artifacts by training the QANet model described in Section 4.3.3 on a version of DROP where either the question or the paragraph input representation vectors are zeroed out (**question-only** and **paragraph-only**, respectively). Consequently, the resulting models can then only predict answer spans from either the question or the paragraph.

In addition, we devise a baseline that estimates the answer variance in DROP. We start by counting the unigram and bigram answer frequency for each wh question-word in the train set (as the first word in the question). The **majority baseline** then predicts an answer as the set of 3 most common answer spans for the input question word (e.g., for “when”, these were “quarter”, “end” and “October”).

4.4 NAQANet

DROP is designed to encourage models that combine neural reading comprehension with symbolic reasoning. None of the baselines we described in Section 4.3.1 can do this. As a preliminary attempt toward this goal, we propose a numerically-aware QANet model, NAQANet, which allows the state-of-the-art reading comprehension system to produce three new answer types: (1) spans from the question; (2) counts; (3) addition or subtraction over numbers. To predict numbers, the model first predicts whether the answer is a count or an arithmetic expression. It then predicts the specific numbers involved in the expression. This can be viewed as the neural model producing a *partially*

executed logical form, leaving the final arithmetic to a symbolic system. While this model can currently only handle a very limited set of operations, we believe this is a promising approach to combining neural methods and symbolic reasoning. The model is trained by marginalizing over all execution paths that lead to the correct answer.

4.4.1 Model Description

Our NAQANet model follows the typical architecture of previous reading comprehension models, which is composed of embedding, encoding, passage-question attention, and output layers. We use the original QANet architecture for everything up to the output layer. This gives us a question representation $\mathbf{Q} \in \mathbb{R}^{m \times d}$, and a projected question-aware passage representation $\bar{\mathbf{P}} \in \mathbb{R}^{n \times d}$. We have four different output layers, for the four different kinds of answers the model can produce:

Passage span As in the original QANet model, to predict an answer in the passage we apply three repetitions of the QANet encoder to the passage representation $\bar{\mathbf{P}}$ and get their outputs as \mathbf{M}_0 , \mathbf{M}_1 , \mathbf{M}_2 respectively. Then the probabilities of the starting and ending positions from the passage can be computed as:

$$\mathbf{p}^{\text{p.start}} = \text{softmax}(\text{FFN}(\mathbf{M}_0; \mathbf{M}_1)) \quad (4.1)$$

$$\mathbf{p}^{\text{p.end}} = \text{softmax}(\text{FFN}(\mathbf{M}_0; \mathbf{M}_2)) \quad (4.2)$$

where FFN is a two-layer feed-forward network with the RELU activation.

Question span Some questions in DROP have their answer in the *question* instead of the passage. To predict an answer from the question, the model first computes a vector \mathbf{h}^P that represents the

information it finds in the passage:

$$\alpha^P = \text{softmax}(\mathbf{W}^P \bar{\mathbf{P}}), \quad (4.3)$$

$$\mathbf{h}^P = \alpha^P \bar{\mathbf{P}} \quad (4.4)$$

Then it computes the probabilities of the starting and ending positions from the question as:

$$\mathbf{p}^{\text{q-start}} = \text{softmax}(\text{FFN}(\mathbf{Q}; \mathbf{e}^{|\mathcal{Q}|} \otimes \mathbf{h}^P), \quad (4.5)$$

$$\mathbf{p}^{\text{q-end}} = \text{softmax}(\text{FFN}(\mathbf{Q}; \mathbf{e}^{|\mathcal{Q}|} \otimes \mathbf{h}^P) \quad (4.6)$$

where the outer product with the identity ($\mathbf{e}^{|\mathcal{Q}|} \otimes \cdot$) simply repeats \mathbf{h}^P for each question word.

Count We model the capability of counting as a multi-class classification problem. Specifically, we consider ten numbers (0–9) in this preliminary model and the probabilities of choosing these numbers is computed based on the passage vector \mathbf{h}^P :

$$\mathbf{p}^{\text{count}} = \text{softmax}(\text{FFN}(\mathbf{h}^P)) \quad (4.7)$$

Arithmetic expression Many questions in DROP require the model to locate multiple numbers in the passage and add or subtract them to get the final answer. To model this process, we first extract all the numbers from the passage and then learn to assign a plus, minus or zero for each number. In this way, we get an arithmetic expression composed of signed numbers, which can be evaluated to give the final answer.

To do this, we first apply another QANet encoder to \mathbf{M}_2 and get a new passage representation \mathbf{M}_3 . Then we select an index over the concatenation of \mathbf{M}_0 and \mathbf{M}_3 , to get a representation for each number in this passage. The i^{th} number can be represented as \mathbf{h}_i^N and the probabilities of this number

In the first quarter, the Saints trailed early as quarterback Ben Roethlisberger completed a 37-yard touchdown pass to WR Hines Ward and a 1-yard touchdown pass to TE Heath Miller. Afterwards, New Orleans got on the board with quarterback Drew Brees completing a 3-yard touchdown pass to WR Terrance Copper ... rookie RB Reggie Bush got a 35-yard touchdown run on a reverse. Pittsburgh would get kicker Jeff Reed to get a 32-yard field goal, yet New Orleans responded with RB Deuce McAllister getting a 29-yard touchdown run. In the third quarter, the Steelers started to fight back as Roethlisberger completed a 38-yard touchdown pass to WR Cedrick Wilson for the only score of the period. In the fourth quarter, RB Willie Parker followed up with a 3-yard and a 4-yard touchdown run to give Pittsburgh the lead.

Question: What was the smallest touchdown run of the game, 1-yard or 3-yard?

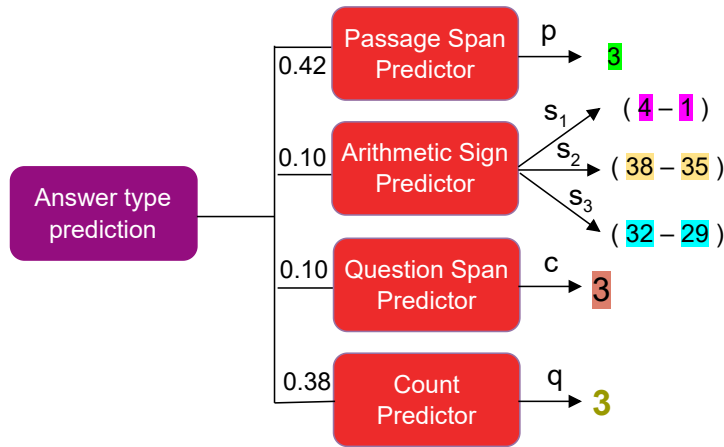


Figure 4.2: Numerically-augmented QANet. The likelihood of answer being “3” is $0.42p + 0.1(s_1 + s_2 + s_3) + 0.10c + 0.38q$

being assigned a plus, minus or zero are:

$$\mathbf{p}_i^{\text{sign}} = \text{softmax}(\text{FFN}(\mathbf{h}_i^N)) \quad (4.8)$$

Answer type prediction We use a categorical variable to decide between the above four answer types, with probabilities computed as:

$$\mathbf{p}^{\text{type}} = \text{softmax}(\text{FFN}(\mathbf{h}^P, \mathbf{h}^Q)) \quad (4.9)$$

where \mathbf{h}^Q is computed over \mathbf{Q} , in a similar way as we did for \mathbf{h}^P . At test time, we first determine this answer type greedily and then get the best answer from the selected type.

4.4.2 Weakly-Supervised Training

For supervision, DROP contains only the answer string, not which of the above answer types is used to arrive at the answer. To train our model, we adopt the weakly supervised training method widely used in the semantic parsing literature [158]. We find all executions that evaluate to the correct answer, including matching passage spans and question spans, correct count numbers, as well as sign assignments for numbers. Our training objective is then to maximize the marginal likelihood of these executions.¹⁰

4.5 Results and Discussion

The performance of all tested models on the DROP dataset is presented in Table 4.4. Most notably, all models perform significantly worse than on other prominent reading comprehension datasets, while human performance remains at similar high levels.¹¹ For example, BERT, the current state-of-the-art on SQuAD, *drops* by more than 50 absolute F1 points. This is a positive indication that DROP is indeed a challenging reading comprehension dataset, which opens the door for tackling new and complex reasoning problems on a large scale.

The best performance is obtained by our NAQANet model. Table 4.6 shows that our gains are obtained on the challenging and frequent number answer type, which requires various complex types of reasoning.

Future work may also try combining our model with BERT. Furthermore, we find that all heuristic baselines do poorly on our data, hopefully attesting to relatively small biases in DROP.

¹⁰Due to the exponential search space and the possible noise, we only search the addition/subtraction of two numbers. Given this limited search space, the search and marginalization are exact.

¹¹Human performance was estimated by the authors collectively answering 560 questions from the test set, which were then evaluated using the same metric as learned systems. This is in contrast to holding out one gold annotation and evaluating it against the other annotations, as done in prior work, which underestimates human performance relative to systems.

Difficulties of building semantic parsers We see that all the semantic parsing baselines perform quite poorly on DROP. This is mainly because of our pipeline of extracting tabular information from paragraphs, followed by the denotation-driven logical form search, can yield logical forms only for a subset of the training data. For SRL and syntactic dependency sentence representation schemes, the search was able to yield logical forms for 34% of the training data, whereas with OpenIE, it was only 25%. On closer examination of a sample of 60 questions and the information extracted by the SRL scheme (the best performing of the three), we found that only 25% of the resulting tables contained information needed to answer the questions. These observations show that high quality information extraction is a strong prerequisite for building semantic parsers for DROP. Additionally, the fact that this is a weakly supervised semantic parsing problem also makes training hard. The biggest challenge in this setup is the spuriousness of logical forms used for training, where the logical form evaluates to the correct denotation but does not actually reflect the semantics of the question. This makes it hard for the model trained on these spurious logical forms to generalize to unseen data. From the set of logical forms for a sample of 60 questions analyzed, we found that only 8 questions (13%) contained non-spurious logical forms.

Error Analysis Finally, in order to better understand the outstanding challenges in DROP, we conducted an error analysis on a random sample of 100 erroneous NAQANet predictions. The most common errors were on questions which required complex type of reasoning, such as arithmetic operations (evident in 51% of the errors), counting (30%), domain knowledge and common sense (23%), co-reference (6%), or a combination of different types of reasoning (40%). See Table 4.5 for examples of some of the common phenomena.

4.6 Conclusion

We have presented DROP, a dataset of complex reading comprehension questions that require **Discrete Reasoning Over Paragraphs**. This dataset is substantially more challenging than existing datasets, with the best baseline achieving only 32.7% F1, while humans achieve 96%. We hope this dataset will spur research into more comprehensive analysis of paragraphs, and into methods that combine distributed representations with symbolic reasoning. We have additionally presented initial work in this direction, with a model that augments QANet with limited numerical reasoning capability, achieving 47% F1 on DROP.

Reasoning	Passage (some parts shortened)	Question	Answer
Subtraction (28.8%)	That year, his Untitled (1981) , a painting of a haloed, black-headed man with a bright red skeletal body, depicted amid the artists signature scrawls, was sold by Robert Lehrman for \$16.3 million, well above its \$12 million high estimate.	How many more dollars was the Untitled (1981) painting sold for than the 12 million dollar estimation?	
Comparison (18.2%)	In 1517, the seventeen-year-old King sailed to Castile. There, his Flemish court In May 1518, Charles traveled to Barcelona in Aragon.	Where did Charles travel to first, Castile or Barcelona?	Castile
Selection (19.4%)	In 1970, to commemorate the 100th anniversary of the founding of Baldwin City, Baker University professor and playwright Don Mueller and Phyllis E. Braun, Business Manager, produced a musical play entitled The Ballad Of Black Jack to tell the story of the events that led up to the battle.	Who was the University professor that helped produce The Ballad Of Black Jack, Ivan Boyd or Don Mueller?	Don Mueller
Addition (11.7%)	Before the UNPROFOR fully deployed, the HV clashed with an armed force of the RSK in the village of Nos Kalik, located in a pink zone near Šibenik, and captured the village at 4:45 p.m. on 2 March 1992. The JNA formed a battlegroup to counterattack the next day.	What date did the JNA form a battlegroup to counterattack after the village of Nos Kalik was captured?	3 March 1992
Count (16.5%) and Sort (11.7%)	Denver would retake the lead with kicker Matt Prater nailing a 43-yard field goal , yet Carolina answered as kicker John Kasay ties the game with a 39-yard field goal. . . . Carolina closed out the half with Kasay nailing a 44-yard field goal. . . . In the fourth quarter, Carolina sealed the win with Kasay's 42-yard field goal.	Which kicker kicked the most field goals?	John Kasay
Coreference Resolu- tion (3.7%)	James Douglas was the second son of Sir George Douglas of Pittendreich, and Elizabeth Douglas, daughter David Douglas of Pittendreich. Before 1543 he married Elizabeth , daughter of James Douglas, 3rd Earl of Morton. In 1553 James Douglas succeeded to the title and estates of his father-in-law.	How many years after he married Elizabeth did James Douglas succeed to the title and estates of his father-in-law?	10
Other Arith- metic (3.2%)	Although the movement initially gathered some 60,000 adherents , the subsequent establishment of the Bulgarian Exarchate reduced their number by some 75%.	How many adherents were left after the establishment of the Bulgarian Exarchate?	15000
Set of spans (6.0%)	According to some sources 363 civilians were killed in Kavadarci , 230 in Negotino and 40 in Vatasha.	What were the 3 villages that people were killed in?	Kavadarci, Negotino, Vatasha
Other (6.8%)	This Annual Financial Report is our principal financial statement of accountability. The AFR gives a comprehensive view of the Department's financial activities ...	What does AFR stand for?	Annual Financial Report

Table 4.1: Example questions and answers from the drop dataset, showing the relevant parts of the associated passage and the reasoning required to answer the question.

Statistic	Train	Dev	Test
Number of passages	5565	582	588
Avg. passage len [words]	213.45	191.62	195.12
Number of questions	77,409	9,536	9,622
Avg. question len [words]	10.79	11.17	11.23
Avg. questions / passage	13.91	16.38	16.36
Question vocabulary size	29,929	8,023	8,007

Table 4.2: Dataset statistics across the different splits.

Answer Type	Percent	Example
NUMBER	66.1	12
PERSON	12.2	Jerry Porter
OTHER	9.4	males
OTHER ENTITIES	7.3	Seahawks
VERB PHRASE	3.5	Tom arrived at Acre
DATE	1.5	3 March 1992

Table 4.3: Distribution of answer types in training set.

Method	Dev		Test	
	EM	F_1	EM	F_1
Heuristic Baselines				
Majority	0.09	1.38	0.07	1.44
Q-only	4.28	8.07	4.18	8.59
P-only	0.13	2.27	0.14	2.26
Semantic Parsing				
Syn Dep	9.38	11.64	8.51	10.84
OpenIE	8.80	11.31	8.53	10.77
SRL	9.28	11.72	8.98	11.45
SQuAD-style RC				
BiDAF	26.06	28.85	24.75	27.49
QANet	27.50	30.44	25.50	28.36
QANet+ELMo	27.71	30.33	27.08	29.67
BERT	30.10	33.36	29.45	32.70
NAQANet				
+ Q Span	25.94	29.17	24.98	28.18
+ Count	30.09	33.92	30.04	32.75
+ Add/Sub	43.07	45.71	40.40	42.96
Complete Model	46.20	49.24	44.07	47.01
Human	-	-	94.09	96.42

Table 4.4: Performance of the different models on our development and test set, in terms of Exact Match (EM), and numerically-focused F_1 (§4.3.1). Both metrics are calculated as the maximum against a set of gold answers.

Phenomenon	Passage Highlights	Question	Answer	Our model
Subtraction + Coreference	... Twenty-five of his 150 men were sick, and his advance stalled ...	How many of Bartolomé de Amésqueta’s 150 men were not sick?	125	145
Count + Filter	... Macedonians were the largest ethnic group in Skopje, with 338,358 inhabitants ... Then came ... Serbs (14,298 inhabitants), Turks (8,595), Bosniaks (7,585) and Vlachs (2,557) ...	How many ethnicities had less than 10000 people?	3	2
Domain knowledge	... Smith was sidelined by a torn pectoral muscle suffered during practice ...	How many quarters did Smith play?	0	2
Addition	... culminating in the Battle of Vienna of 1683, which marked the start of the 15-year-long Great Turkish War ...	What year did the Great Turkish War end?	1698	1668

Table 4.5: Representative examples from our model’s error analysis. We list the identified semantic phenomenon, the relevant passage highlights, a gold question-answer pair, and the erroneous prediction by our model.

Type	(%)	Exact Match		F1	
		QN+	BERT	QN+	BERT
Date	1.57	28.7	38.7	35.5	42.8
Numbers	61.94	44.0	14.5	44.2	14.8
Single Span	31.71	58.2	64.6	64.6	70.1
> 1 Spans	4.77	0	0	17.13	25.0

Table 4.6: Dev set performance breakdown by different answer types; our model (NAQANet, marked as *QN+*) vs. BERT, the best-performing baseline.

Passage

After the deaths of Charles V and du Guesclin in 1380, France lost its main leadership and overall momentum in the war. Charles VI succeeded his father as king of France at the age of 11, and he was thus put under a regency led by his uncles, who managed to maintain an effective grip on government affairs until about 1388, well after Charles had achieved royal majority. With France facing widespread destruction, plague, and economic recession, high taxation put a heavy burden on the French peasantry and urban communities. The war effort against England largely depended on royal taxation, but the population was increasingly unwilling to pay for it, as would be demonstrated at the Harelle and Maillotin revolts in 1382. Charles V had abolished many of these taxes on his deathbed, but subsequent attempts to reinstate them stirred up hostility between the French government and populace. Difficulties in raising taxes and revenue hampered the ability of the French to fight the English. At this point, the war's pace had largely slowed down, and both nations found themselves fighting mainly through proxy wars, such as during the 1383-1385 Portuguese interregnum. The independence party in the Kingdom of Portugal, which was supported by the English, won against the supporters of the King of Castile's claim to the Portuguese throne, who in turn was backed by the French.

Type a question based on the passage below

When was Charles VI born?

Your Answer

Date

Year	Month	Date
1368		

Number

Select span

AI predicted the answer below (wait for answer to appear below)

11

Q: For how many years did Charles VI's uncle maintain the government affairs?
A: 8

ADD QUESTION

READY TO SUBMIT HIT

PREVIOUS PASSAGE

NEXT PASSAGE

Question: Which team had the longest touchdown pass?

Still searching for their first win, the Bengals flew to Texas Stadium for a Week 5 interconference duel with the Dallas Cowboys. In the first quarter, Cincinnati trailed early as Cowboys kicker Nick Folk got a 30-yard field goal, along with RB Felix Jones getting a 33-yard TD run. In the second quarter, Dallas increased its lead as QB Tony Romo completed a 4-yard TD pass to TE Jason Witten. The Bengals would end the half with kicker Shayne Graham getting a 41-yard and a 31-yard field goal. In the third quarter, Cincinnati tried to rally as QB Carson Palmer completed an 18-yard TD pass to WR T. J. Houshmandzadeh. In the fourth quarter, the Bengals got closer as Graham got a 40-yard field goal, yet the Cowboys answered with Romo completing a 57-yard TD pass to WR Terrell Owens. Cincinnati tried to come back as Palmer completed a 10-yard TD pass to Houshmandzadeh (with a failed 2-point conversion), but Dallas pulled away with Romo completing a 15-yard TD pass to WR Patrick Crayton.

Which Bengals receiver scored two touchdowns?

Still searching for their first win, the Bengals flew to Texas Stadium for a Week 5 interconference duel with the Dallas Cowboys. In the first quarter, Cincinnati trailed early as Cowboys kicker Nick Folk got a 30-yard field goal, along with RB Felix Jones getting a 33-yard TD run. In the second quarter, Dallas increased its lead as QB Tony Romo completed a 4-yard TD pass to TE Jason Witten. The Bengals would end the half with kicker Shayne Graham getting a 41-yard and a 31-yard field goal. In the third quarter, Cincinnati tried to rally as QB Carson Palmer completed an 18-yard TD pass to WR T. J. Houshmandzadeh. In the fourth quarter, the Bengals got closer as Graham got a 40-yard field goal, yet the Cowboys answered with Romo completing a 57-yard TD pass to WR Terrell Owens. Cincinnati tried to come back as Palmer completed a 10-yard TD pass to Houshmandzadeh (with a failed 2-point conversion), but Dallas pulled away with Romo completing a 15-yard TD pass to WR Patrick Crayton.

Which alliance lost more troops to prisoner status?

About eight million men surrendered and were held in POW camps during the war. All nations pledged to follow the Hague Conventions on fair treatment of prisoners of war, and the survival rate for POWs was generally much higher than that of combatants at the front. Individual surrenders were uncommon; large units usually surrendered en masse. At the siege of Maubeuge about 40,000 French soldiers surrendered, at the battle of Galicia Russians took about 100,000 to 120,000 Austrian captives, at the Brusilov Offensive about 325,000 to 417,000 Germans and Austrians surrendered to Russians, and at the Battle of Tannenberg 92,000 Russians surrendered. When the besieged garrison of Kaunas surrendered in 1915, some 20,000 Russians became prisoners, at the battle near Przasnysz 14,000 Germans surrendered to Russians, and at the First Battle of the Marne about 12,000 Germans surrendered to the Allies. 25-31% of Russian losses were to prisoner status; for Austria-Hungary 32%, for Italy 26%, for France 12%, for Germany 9%; for Britain 7%. Prisoners from the Allied armies totalled about 1.4 million. From the Central Powers about 3.3 million men became prisoners; most of them surrendered to Russians. Germany held 2.5 million prisoners; Russia held 2.2-2.9 million; while Britain and France held about 720,000. Most were captured just before the Armistice. The United States held 48,000. The most dangerous moment was the act of surrender, when helpless soldiers were sometimes gunned down. Once prisoners reached a camp, conditions were, in general, satisfactory, thanks in part to the efforts of the International Red Cross and inspections by neutral nations. However, conditions were terrible in Russia: starvation was common for prisoners and civilians alike; about 15-20% of the prisoners in Russia died, and in Central Powers imprisonment 8% of Russians. In Germany, food was scarce, but only 5% died.

How many years after the first Kandyan War did the second Kandyan War happen?

During the Napoleonic Wars, Great Britain, fearing that French control of the Netherlands might deliver Sri Lanka to the French, occupied the coastal areas of the island with little difficulty in 1796. In 1802, the Treaty of Amiens formally ceded the Dutch part of the island to Britain and it became a crown colony. In 1803, the British invaded the Kingdom of Kandy in the first Kandyan War, but were repulsed. In 1815 Kandy was occupied in the second Kandyan War, finally ending Sri Lankan independence. Following the suppression of the Uva Rebellion the Kandyan peasantry were stripped of their lands by the Wastelands Ordinance, a modern enclosure movement, and reduced to penury. The British found that the uplands of Sri Lanka were very suitable for coffee, tea and rubber cultivation. By the mid-19th century, Ceylon tea had become a staple of the British market bringing great wealth to a small number of white tea planters. The planters imported large numbers of Tamil workers as indentured labourers from south India to work the estates, who soon made up 10% of the island's population. These workers had to work in slave-like conditions living in line rooms, not very different from cattle sheds.

Figure 4.3: Question Answering HIT sample above with passage on the left and input fields for answer on the right and Highlighted candidate spans of sample answers below

Chapter 5

Successive Prompting

If there is a lesson to be drawn from the history of cognitive science, it is that the most important discoveries arise when we break the world into small pieces.

Jerry Fodor

(Modularity of Mind - 1983)

Answering complex questions that require making latent decisions is a challenging task, especially when limited supervision is available. Recent works leverage the capabilities of large language models (LMs) to perform complex question answering in a few-shot setting by demonstrating how to output intermediate rationalizations while solving the complex question in a single pass. We introduce “Successive Prompting”, where we iteratively break down a complex task into a simple task, solve it, and then repeat the process until we get the final solution. Successive prompting decouples the supervision for decomposing complex questions from the supervision for answering simple questions, allowing us to (1) have multiple opportunities to query in-context examples at each reasoning step (2) learn question decomposition separately from question answering, including

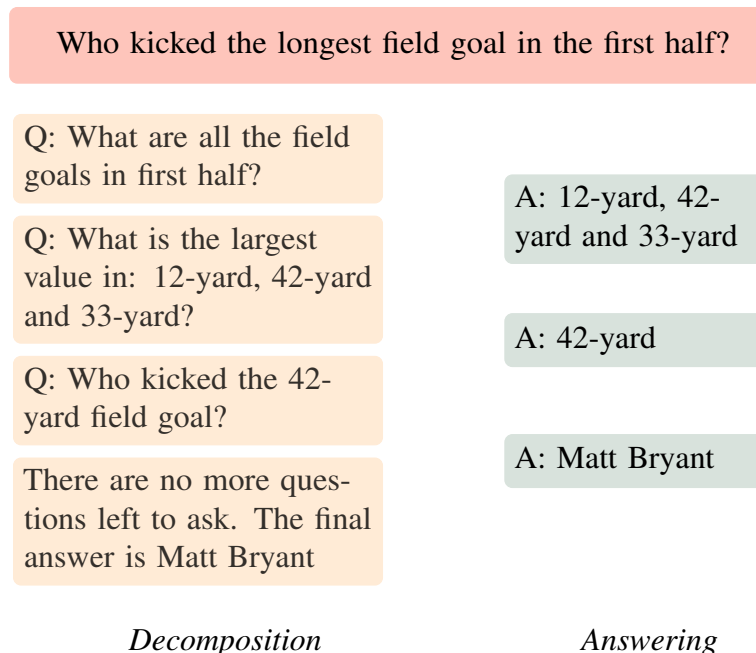


Figure 5.1: Example decomposition used by Successive Prompting’s question decomposition and question answering stage on a DROP example. The model iterates between predicting a simple question to ask and answering the simple question.

using synthetic data, and (3) use bespoke (fine-tuned) components for reasoning steps where a large LM does not perform well. The intermediate supervision is typically manually written, which can be expensive to collect. We introduce a way to generate a synthetic dataset which can be used to bootstrap a model’s ability to decompose and answer intermediate questions. Our best model (with successive prompting) achieves an improvement of $\sim 5\%$ absolute F1 on a few-shot version of the DROP dataset when compared with a state-of-the-art model with the same supervision.

5.1 Introduction

Compositional reading comprehension datasets like HotpotQA [40] and DROP [17] have inspired a range of model architectures that learn to answer complex questions with weak supervision from the final answer. One recent direction is to leverage large language models (LMs) to solve compositional tasks with very few examples by generating latent reasoning steps before answering

the question [98, 172, 173].

Given a complex question, this approach first finds nearest-neighbor training examples from a dataset of (question, reasoning, answer) triples and then concatenates them to create an input for the LM. A large LM is then prompted with this input to generate the intermediate reasoning steps needed, while answering the complex question in a single pass.

While promising, this approach discards many of the benefits of prior approaches to this task [174, 173] by coupling the supervision for question decomposition to the supervision for performing the intermediate steps. Moreover, its non-modular nature does not allow using alternate symbolic reasoning engines in cases where they perform better than LMs. Additionally, the model gets exposed to only a single set of in-context examples, selected based on their proximity to the complex question, which may not contain optimal supervision for the intermediate steps that need to be taken.

We propose “Successive Prompting”, where we iteratively decompose the complex question into the next simple question to answer, answer it, and then repeat until the complex question is answered (Figure 5.1). Each of these steps is performed with separate a query to the LM. Since the decomposition and answering steps are performed separately, we can decouple the supervision of each step, providing two primary benefits. First, when performing in-context learning, we get multiple opportunities to select different in-context examples, which can be tailored to the particular decomposition or answering step being performed, instead of selecting a single set of examples based only on the complex question. Second, when fine-tuning (with or without in-context examples [175]), we can provide training examples for each step independently, so the model only has to learn to perform one step at a time.

This decoupling additionally allows us to judiciously inject synthetic data into the learning process, e.g., to help the model answer a particular kind of simple question that it could not previously answer, or a new reasoning composition it did not know how to decompose. Because the steps are

separate, we can isolate model failures and develop synthetic approaches to fill in the gaps. It also allows us to replace the LM with other, purpose-built components to perform symbolic reasoning when appropriate [174, 176, 177].

We demonstrate the utility of successive prompting using a few-shot variant of the DROP dataset [17], selecting 300 examples for training (either fine-tuning or in-context example selection). These 300 examples are manually annotated with simple QA pairs as decompositions. We find that performance of all models is quite low in this few-shot setting, so we develop a synthetic data generator that produces complex questions with their decompositions from semi-structured Wikipedia tables [95]. This synthetic data provides not just complex question supervision, but also supervision for the intermediate steps. We augment this data with the 300 (complex) training examples and their decompositions from DROP. In this few-shot setting, our best performing successive prompting model shows a $\sim 5\%$ improvement in F1 when compared to state-of-the-art model on DROP.

5.2 Decomposing Complex Questions

The goal of compositional question answering is to answer a complex question q in the context of a passage p (together denoted as x) by reasoning through latent sequential decisions $\mathbf{z} = z_1, z_2, \dots, z_s$ to reach the final answer, y . Many models have been proposed to accomplish this with varying amounts of supervision and interpretability. In prompting methods like Chain-of-Thought [CoT, 98] the latent steps are supervised, interpretable sentences; in other models these latent steps might be a program [178, 179] or even just the (unsupervised) hidden states in the model [176, 180]

We focus on models that take in-context examples and produce a discrete, language-encoded \mathbf{z} , with CoT being the primary exemplar. We write the general form for CoT, given an input x , a language model encoder \mathbb{L} and N in-context examples obtained from querying an index \mathcal{I} —each containing a triplet of passage with complex question (x^n), latent steps (\mathbf{z}^n) and final answer (y^n)—as follows:

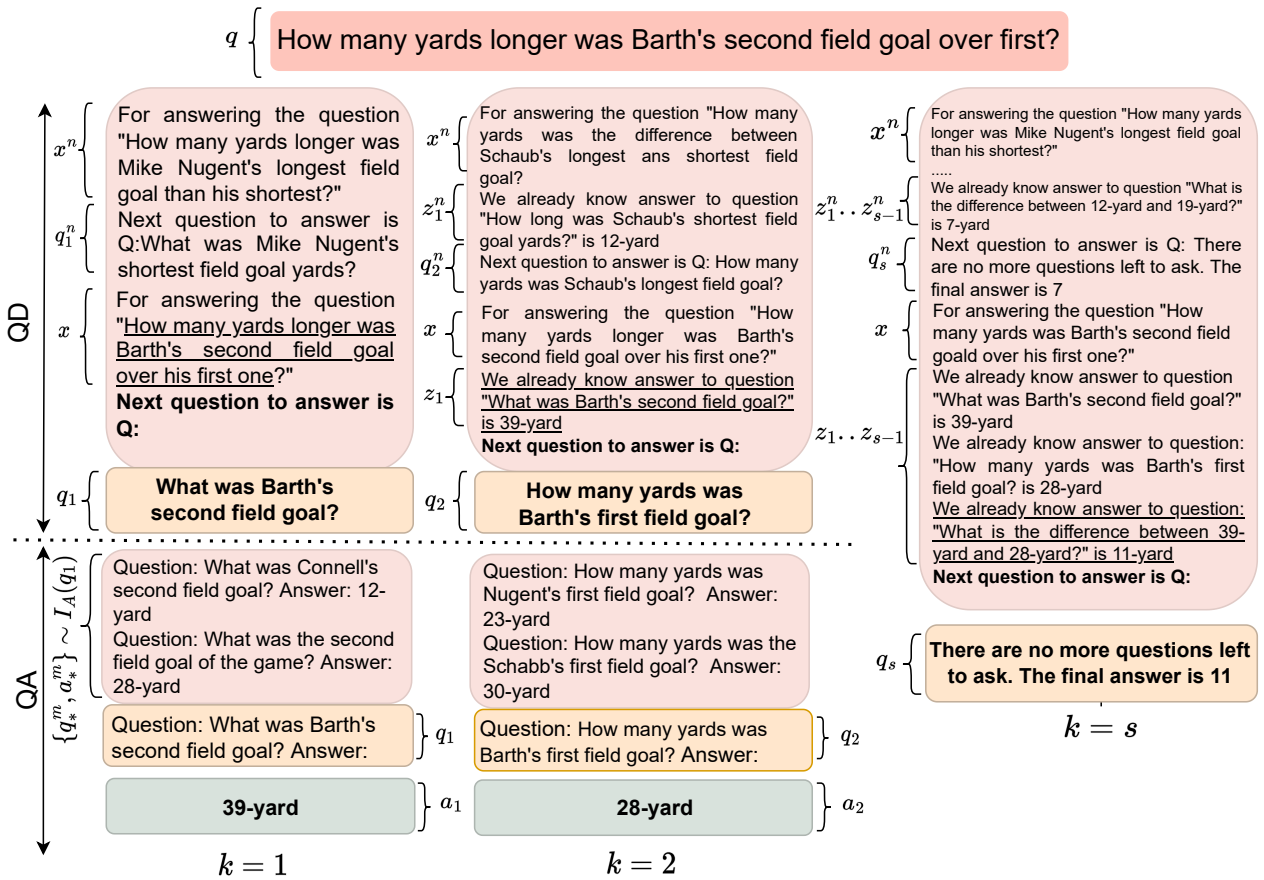


Figure 5.2: A demonstration of successive prompting with in-context learning. The selected examples for supervision and complex question to be answered pre-pended with the context paragraph (omitted to simplify illustration) are encoded by the model to generate question and answer at QD and QA stage respectively. During fine-tuning, only training supervision is used in an i.i.d manner for learning QD and QA models.

$$y, z \leftarrow \mathbb{L}(x, \{(x^n, y^n, \mathbf{z}^n) \mid n \in [1, N]\})$$

5.2.1 Successive prompting

In successive prompting, we represent each latent step as a pair of simple question and answer, $z_k = (q_k, a_k)$ (see Figure 5.1 for example QA pairs) unlike CoT which represents each latent step as a declarative sentence. Moreover, CoT queries the index \mathcal{I} for in-context examples and prompts the language model \mathbb{L} for generating output only once. However, in successive prompting, we separate \mathbf{z} into multiple question and answering steps, which gives us many opportunities to prompt \mathbb{L} , with potentially different in-context examples that are more tailored to the simple question at each step. It also enables us to re-encode the context given the intermediate state z_k , which can be useful in certain questions that need long chain referencing (e.g., the sort-count example in Figure 5.3). We can write a general form for successive prompting as follows:

$$\begin{aligned} q_1 &\leftarrow \mathbb{L}(x, \{(x^n, q_1^n) \mid n \in [1, N]\}) \\ a_1 &\leftarrow \mathbb{L}(p, q_1, \{(p_*^m, q_*^m, a_*^m) \mid m \in [1, M]\}) \\ q_2 &\leftarrow \mathbb{L}(x, q_1, a_1, \{(x^n, q_1^n, a_1^n, q_2^n) \mid n \in [1, N]\}) \\ a_2 &\leftarrow \mathbb{L}(p, q_2, \{(p_*^m, q_*^m, a_*^m) \mid m \in [1, M]\}) \\ &\dots \\ y &\leftarrow \mathbb{L}(x, \mathbf{z}, \{(x^n, y^n, \mathbf{z}^n) \mid n \in [1, N]\}) \end{aligned}$$

There are three kinds of model outputs in this general form: intermediate questions q_k , intermediate answers a_k , and the final answer y . We refer to the first kind of output as *question decomposition*

(QD) and the second kind as *question answering* (QA). We treat final answer prediction as a special case of question decomposition, where the model decides that no more decomposition is necessary and outputs a final answer, so we iteratively alternate between question decomposition and question answering until the model terminates.

5.2.2 Training paradigm

We have so far described successive prompting in a setting where only in-context examples are given, so no model training is performed. However, successive prompting can also be used in conjunction with model fine-tuning, where each intermediate output is treated as a training example for \mathbb{L} . In this section, we first describe how in-context examples are selected at every step, followed by detailing how these examples are used for model fine-tuning.

In-context Learning During in-context learning, a small number of training examples are provided directly in the prompt that is given to a large LM, before the test input. These examples are selected from an index based on their similarity with the test input. For successive prompting, we create two indices: \mathcal{I}_D , for looking-up relevant demonstrations for QD, and \mathcal{I}_A , for looking-up relevant demonstrations for QA. The index \mathcal{I}_D contains partially decomposed chains at each step k , demonstrating the next question q_k to be produced for every complex question in the training data. The index \mathcal{I}_A contains all the simple QA pairs in the training data from all the complex questions.

In the QD stage, the index \mathcal{I}_D is queried with the complex test question, q and current step number, k , to select demonstrations regarding how to generate the next question for the held-out example. In the QA stage, the index \mathcal{I}_A is queried with the simple question q_k generated during QD to select relevant simple QA pairs. Figure 5.2 shows a demonstration of how in-context learning is executed step-by-step in each stage until QD outputs the special phrase “There are no more questions left to ask”, along with a final answer.

Successive prompting allows the QA stage to access simple questions derived from complex questions that would not have been retrieved by Chain-of-Thought prompting because on the surface they are not similar to the held-out complex question, even though they share similar sub-questions.

Model Fine-tuning For model fine-tuning, we use T5 [33] based sequence-to-sequence models. Such models are typically trained with control codes in a multi-task setting [181, 182] to switch between QD and QA tasks with shared model parameters. We adapt and extend the control codes introduced by text modular networks [TMNs, 174] for training with our synthetic data. TMNs are limited in terms of the operations they can handle as they do not go beyond first order reasoning. We use synthetically generated data, which allows us to deal with higher-order reasoning questions in DROP. Because we are fine-tuning the model, we can use special tokens to denote question decomposition and other separators, instead of the natural language prompts shown in Figure 5.2, though the content is the same. The specific tokens used for each step are listed in Appendix A.

Specialized Modules Successive prompting also allows us to use specialized sub-modules for solving different QA tasks because we no longer perform QD and QA in an end-to-end manner. Solving arithmetic operations like counting, difference, sorting, etc., can be challenging for language models. As a result, we follow [174] and construct a simple mathematical sub-module for QA which parses the generated simple question for symbolic operation type and its arguments and then executes them in a deterministic way. If the generated simple question cannot be parsed as a mathematical operation, we apply the language model to solve it.

5.3 Synthetic Dataset

Any method that prompts LMs to produce intermediate reasoning steps to answer complex questions needs some amount of supervision for those reasoning steps. This kind of annotation can be

expensive to collect and often requires expert knowledge. Prior work has typically relied on a small handful of manually-written example decompositions. We find that such small collections lead to very poor performance on a dataset as varied as DROP, even for large models.

To mitigate these data issues, we propose a way to synthetically generate complex questions and their decompositions using semi-structured data which is easy to parse. We show that we can bootstrap model learning with this out-of-domain, synthetically generated data so it can adapt better when fine-tuned with limited in-domain supervision.

Generation Process: Inspired by [95], we use semi-structured data from tables in English Wikipedia which are available in plenty.

We employ curated templates to convert the rows in the tables into paragraphs. We use single column headers to create first order simple questions and a combination of columns for higher order complex questions.

We synthesize data for 10 simple operations: COUNT, TOP(k), BOTTOM(k), FILTER, SUM, COMPARISON, DIFFERENCE, NEGATION, GATHER, and INTERSECTION.

We generate higher order combinations of first-order operations, wherever possible. Figure 5.3 shows examples of higher order combinations of the atomic operation COUNT with a few other simple operations using Table 5.1 as context. The complete list of all decompositions is provided in Appendix A. Depending on the model, we use either symbolic or natural language version of the arithmetic operations. If we are using an LM to perform arithmetic operations, we output natural language; if we are using a separate symbolic reasoning engine, we output symbolic operations. We generate approximately 141K total complex questions which result in 525K examples for QD and 257K examples for QA. See Appendix A for more dataset statistics.

Reasoning	Complex Question and Decomposition (Question [Natural Language or Symbolic], Answer)
Count	<p>How many opponents were there?</p> <ul style="list-style-type: none"> • Q: What are all the opponents? Ans: Walsall; Liverpool; Sheffield Wed.; Oxford United; Portsmouth • Q: count(Walsall; Portsmouth; Sheffield Wed.; Oxford United; Portsmouth) Ans: 5 <ul style="list-style-type: none"> – Q: How many items are in the list: Walsall, Liverpool, Sheffield Wed. and Oxford United, Portsmouth?
Higher order decompositions	
Sort-Count	<p>Which venue had the most number of opponents?</p> <ul style="list-style-type: none"> • Q: What are all the venues? Ans: A; H • Q: What are opponents when venue was A? Ans: Walsall; Oxford United; Portsmouth • Q: count(Walsall; Oxford United; Portsmouth) Ans: 3 • Q: What are opponents when venue was H? Ans: Liverpool; Sheffield Wed. • Q: count(Liverpool; Sheffield Wed.) Ans: 2 • Q: top(1, 2;3) Ans: 3 <ul style="list-style-type: none"> – Q: What is the largest value in: 2 and 3? • Q: Which venue has 3 opponents? Ans: A
Comparison-Count	<p>Which round had more venues: SF 1st Leg or QFR?</p> <ul style="list-style-type: none"> • Q: What are the rounds when venue was A? Ans: R2 1st Left; SF 2nd Leg; QFR • count(R2 1st Left; SF 2nd Leg; QFR) Ans: 3 • Q: What are the rounds when venue was H? Ans: QFR; SF 1st Leg • count(QFR; SF 1st Leg) Ans: 2 • if_then(1 > 2; SF 1st Leg; QFR) Ans: QFR <ul style="list-style-type: none"> – Q: If 1 > 2 then answer is SF 1st Leg else it is QFR

Figure 5.3: Examples of COUNT operation and some of its higher order combinations, with natural language and symbolic decompositions of the complex question. Underneath the first instance of a symbolic operation we show its corresponding natural language version. See Table 5.1 for the original table used to generate context and questions.

Round	Date	Opponent	Venue	Attendance
R2 1st Leg	26 Sep 1990	Walsall	A	5,666
QFR	23 Oct 1990	Liverpool	H	18,246
SF 1st Leg	24 Feb 1991	Sheffield Wed.	H	14,074
SF 2nd Leg	27 Feb 1991	Oxford United	A	34,669
QFR	23 Jan 1991	Portsmouth	A	33,861

Table 5.1: Example table from Wikipedia where rows become sentences and columns are used for question generation (used as context for Figure 5.3).

5.4 Experiments and Results

The DROP dataset contains a variety of reasoning compositions which are not uniformly distributed. In order to get a fair representation of DROP examples, we first embed the examples using a sentence embedding method trained on the QQP dataset [183]. We then use cosine similarity to get the top-50 nearest neighbor questions for each training example. The connection graph between each training question to its neighbors is then used to obtain 300 questions that cover the majority of the training data, via the vertex cover algorithm. We manually annotate these 300 examples with decomposed QA pairs in the same format as our synthetic data (Figure 5.3). For synthetic examples, since we know the reasoning types, we uniformly sample example demonstration from each reasoning type.

5.4.1 In-context Learning

Setup We use faiss¹ index with the QQP-based sentence embedding [183] for indexing all the questions. We use GPT-J (6B)² which is the largest freely available model we could use with prompts containing 6 in-context examples.

¹<https://ai.facebook.com/tools/faiss/>

²<https://github.com/EleutherAI>

	Syn-Only	DROP-Only	Syn+ DROP
Standard	22.7	23.8	24.9
CoT	25.3	26.2	27.6
Succ.(w/o calc.)	27.2	29.3	29.9
Succ.(w/ calc.)	28.8	30.8	31.9

Table 5.2: F1 Performance of in-context prompting on the DROP dev set with and without in-domain annotations.

Results In Table 5.2, we compare performance of language models without any prompting (Standard), with chain-of-thought prompting (CoT) and successive prompting. We observe that successive prompting performs better than CoT by 3.5% when only synthetic data is available, and 4.3% better with synthetic data and 300 annotations from DROP. The best successive prompting version on the dev set (Synthetic+DROP) has a test set performance of 30.6% F1. We also perform an ablation where the symbolic calculator is replaced by language model and observe that the performance drops by 1.5% F1. This further shows that modular approach is better over a single model that tries to solve all the tasks.

5.4.2 Model Fine-tuning

Setup We employ a shared question decomposition (QD) and answering model (QA) based on T5-large version of UnifiedQA [74], trained in a multi-task manner. We use the format described in Appendix A for prompting UnifiedQA. For symbolic questions, we use a simple calculator that parses the operator and arguments in the generated question and executes the discrete operator on the detected arguments.

To deter the model from learning incorrect steps, we use contrastive estimation [104]. In particular, we first train the model for two epochs with cross-entropy loss while generating the output sequence (simple question or answer). Then we continue training by adding an auxiliary loss term which increases the likelihood of the intermediate sub-question that would produce a correct sub-answer

at the cost of one that does not [9]. We sample up to 3 negative samples at each step. We use HuggingFace transformers³ to train our models, with a learning rate of 5e-5 and maximum input length of 768.

Due to variance in the types of context tables present in Wikipedia, the synthetic dataset distribution is not uniform across different reasoning types. To have a balanced representation of questions pertaining to different reasoning types, we employ dynamic sampling [14], where at the beginning of each epoch we select 80,000 instances from across all reasoning types in proportion to the drop in their current performance with respect to previous epoch on held-out synthetic data. For the first epoch we sample in proportion to original the size of each reasoning type. During inference, we use beam search with size 5 to generate decompositions, switching between QD and QA stages until QD reaches end of decomposition (“EOQ”) or maximum number of steps which we set as 10.

Baseline models We compare against a number of different baselines, both symbolic and non-symbolic. As non-symbolic baselines, we use UnifiedQA [74], which is pre-trained on a number of existing question answering datasets, and PReasM [95], which is pre-trained on synthetically generated compositional QA pairs. We also include a baseline with symbolic components, TASE [176]. This model (and others like it [177, 180]) are capable of performing a combination of continuous and discrete operations, which is essential for DROP. TASE does not require expressing decomposition in a specific grammar and can work with natural language. We chose this model as it is close to state of the art on the full DROP dataset and has publicly available code.

Results In Table 5.3, we use the DROP dev set to compare the performance of different symbolic and non-symbolic models in three settings: (1) using no training data from DROP (0-shot), (2) using only question-answer supervision from the 300 DROP examples, and (3) using both question-answer supervision and the decompositions for the 300 DROP examples. In each of these settings, we can

³<https://github.com/huggingface/transformers>

	0-shot	w/o decomp	w/ decomp
Non-symbolic			
<i>UnifiedQA</i>	24.5	26.7	27.2
+ Synthetic	26.6	30.3	32.6
<i>PReasM</i>	24.9	34.6	37.5
+ Synthetic	30.2	36.2	38.1
Symbolic			
<i>TASE</i>	-	26.1	27.6
+ Synthetic	27.3	44.1	45.9
<i>Succ. Prompting</i>	49.8	-	51.3

Table 5.3: F1 Performance of various model architectures on DROP dev-set pre-trained on synthetic data and further fine-tuned with 300 DROP examples.

train the model with or without the synthetic data that we generated.

We observe that our out-of-domain synthetic data universally improves model performance, and the improvement is most pronounced in TASE, nearing a 20% absolute improvement. Without synthetic data, PReasM is the best performing baseline, but TASE overtakes PReasM when synthetic data is available. Additionally, and unsurprisingly, increasing the amount of supervision from 0-shot to complex QA pairs to decompositions universally improves model performance.

Finally, our method, which is a fine-tuned successive prompting model combined with a symbolic reasoning engine, achieves the best performance, giving an improvement of 5.4 F1 over the state-of-the-art model with similar supervision, i.e. TASE+Synthetic w/ decomp.

We follow the standard practice of using test set for only our final best performing model (SP w/ decomp). We observe that our best model with a test set performance of 50.2 F1 is better than the state-of-the-art model with similar supervision (45.1 F1) by 5.1% F1.

Overall, methods that learn to decompose complex questions into simple QA pairs adapt well to complex questions in new domain with little (SP w/ decomp: 51.3 F1) to no in-domain supervision for decomposition (SP 0-shot: 49.8). If we have limited complex QA supervision (without any decompositions), un-interpretable symbolic models result in the best performance (TASE + Synthetic

	QA: In-Context	QA: Fine-tuning
QD: In-Context	30.8	40.3
QD: Fine-tuning	31.4	51.3

Table 5.4: F1 with QD and QA modules from incontext learning and fine-tuning with only DROP annotations

w/o decomp: 44.1). This is because of two reasons. First, such models can capture domain specific answer priors which may result in decent held-out performance [13, 184]. Second, depending on the context, sometimes it may not be straight-forward to decompose the complex questions into QA pairs.

5.4.3 In-context vs Fine-Tuning

To understand the gap in performance between successive prompting with in-context learning and fine-tuning, we perform ablations across in-context and fine-tuned version of QD and QA modules. We observe that in-context learning is unable to do well on answering simple questions that result in a list of answers—which is especially important for DROP as symbolic aggregations are generally applied on a list of answers. On using a fine-tuned QA model we see an improvement of $\sim 10\%$ in F1 with an in-context QD model. Moreover, since the final answer performance is dependent on how well the QA model performs, using a better QD model (fine-tuned) does not help the overall performance much unless the QA model can handle the decompositions produced by the QD model.

5.4.4 Qualitative Examples

To evaluate the correctness of decomposed QA pairs, we manually analyze a subset of predictions on the dev set with in-context (DROP-only) learning and model fine tuning (few shot). We do this by randomly sampling 50 correct predictions to determine how often the incorrect decompositions result in correct answer. We observe that QD stage has an accuracy of 88% for in-context and 96%

Example	Chain-of-Thought	Successive (In-context)	Successive (Fine-Tuning)
<p>Who threw the longest touchdown pass?</p>	<p>The touchdown passes are 22-yard and eight-yard. Out of 22 and eight-yard, the longest touchdown pass is 22. Peyton manning threw the 22-yard touchdown pass. The answer is Peyton Manning. <i>Answer (✓)</i></p>	<ul style="list-style-type: none"> • What are all the touchdown passes? A: 22-yard, eight-yard • What is the largest value in: 22-yard, eight-yard? A: 22-yard • Who threw the 22-yard touchdown pass? A: Peyton Manning. • There are no more questions left to ask. The final answer is Peyton Manning. <i>Answer (✓)</i> 	<ul style="list-style-type: none"> • What are all the touchdown passes? A: 22-yard, eight-yard • max(22;eight) A: 22 • Who threw the 22-yard touchdown pass? A: Peyton Manning. <i>Answer (✓)</i>
<p>How many field goals did both teams kick in the first half?</p>	<p>The field goals in the game are 20, 24, 53, 30 and 40. The count becomes 7. <i>Answer (✗)</i></p>	<ul style="list-style-type: none"> • What are the field goals both teams kicked in the first half? A: 20, 24 and 53 • How many items are in the list 20, 24 and 53? A: 3 • There are no more questions left to ask. The final answer is 3. <i>Answer (✓)</i> 	<ul style="list-style-type: none"> • What are all the field goals in the first half? A: 20; 24; 53 • count(20; 24; 53) A: 3 <i>Answer (✓)</i>
<p>How many of the personnel were not civil servants?</p>	<p>The personnel are 1,513. Since 1,513 is less than 3,183. The answer is not civil servants. <i>Answer (✗)</i></p>	<ul style="list-style-type: none"> • Q: How many of the personnel were not civil servants? A: 1,176 • Q: What is the difference between 100 and 1176? A: 1111 • There are no more questions left to ask. The final answer is 1111. <i>Answer (✗)</i> 	<ul style="list-style-type: none"> • Q: How many personnel were there? A: 2,234 • Q: How many of the personnel were civil servants? A: 1,176 • Q: diff(2234; 1176) A: 1058 <i>Answer (✓)</i>
<p>Q: Which port did the Korean immigrants leave first Chemulpo or Veracruz? Passage: In 1905, Korean immigrants departed from the port of Chemulpo.. After which they took a train to Veracruz. In the Veracruz port, another boat was taken to the port of Progreso.</p>	<p>The Korean immigrants left Chemulpo. The answer is Chemulpo. Reasoning (✗) Answer (✓)</p>	<ul style="list-style-type: none"> • Q: In what year did the Korean immigrants leave Chemulpo? A: 1905 • Q: Which port did the Korean immigrants leave first Chemulpo or Veracruz? A: Chemulpo • Q: If Chemulpo is greater than Veracruz, answer is Chemulpo else it is Veracruz A: Chemulpo. Reasoning (✗) Answer (✓) 	<ul style="list-style-type: none"> • Q: When did Korean immigrants leave Chemulpo? A: 1905 • Q: When did Korean immigrants leave Veracruz? A: 1905 • Q: if.then(1905 < 1905; Chemulpo; Veracruz) A: Veracruz. <i>Answer (✗)</i>

Figure 5.4: Generated decompositions depicting strength and weaknesses of Successive Prompting.

for fine-tuned model. The incorrect decompositions are mainly because the decomposed question is identical to the original question. For instance, "Who made the longest field goal?" can sometimes be answered correctly without decomposing the question if the passage contains a single field goal mention.

We also sample 50 incorrect predictions to ascertain the reason for incorrect predictions in both in-context and fine-tune setup. We observe that the final predictions are incorrect due to three main categories of errors: incorrect QA model prediction, incorrect next question prediction (QD) and out-of-scope reasoning type. The QA model outputs incorrect answers to simple question 40% and 22% of the times for in-context and fine-tuned respectively. The second class of errors, due to incorrect decomposition, occur 30% of the times for both in-context and fine-tuned. The final class of errors, due to compositional questions that are not covered by synthetically generated annotations, occur 28% (in-context) and 46% (fine-tune) of the times.

In Figure 5.4, we show a few examples of correct and incorrect predictions and point out the strengths and weaknesses of successive prompting. The main strength of successive prompting is that, by breaking down the question, we are able to get improved supervision for QA. As a result, it is able to correctly identify the goals kicked in the first half while answering the question "How many field goals did both teams kick in the first half?", unlike CoT that returns goals for the entire game.

One of the limitations of in-context learning, when compared with fine-tuning (irrespective of the type of prompting), is that examples are chosen based on the question alone, overlooking the context. For instance, DROP has questions like "How many people were not Germans, in terms of percentage?" where we first need to answer "How many people were Germans, in terms of percentage?" and then perform a negation operation (i.e, subtract from 100). The word "not" influences the example lookup to choose decomposition that involves a negation even when the question being answered requires a different operation.

A limitation of successive prompting is that it is sometimes challenging to decompose a question, especially when it involves implicit reasoning from the passage. For instance, for “Which port did the Korean immigrants leave first Chemulpo or Veracruz?”, it is difficult to explicitly define a comparison style decomposition from the sentence, “After which they took a train to Veracruz”.

5.5 Related Work

Prompting methods Prompting was introduced as a way to test the reasoning capabilities of large language models [42]. In follow-up works [185, 96, 186] prompting techniques have been used as a mechanism to supervise the model decision with few demonstrations as a conditioning context to guide its predictions on an unseen example. Works like Chain-of-Thought reasoning [98, 187] especially focus on compositional questions where they provide a chain of reasoning as demonstrations. In concurrent work, Least-to-Most prompting [188] takes a similar view as ours to break down the problem into sub-problems. However, in Successive Prompting the question decomposition and answering stages are interleaved, unlike Least-to-Most where the problem is first reduced into sub-problem and then executed in a sequence. In our method, the next question prediction has access to previously answered sub-questions, which is useful in questions that need long chain referencing. Other contemporaneous works [189, 190] use very large language models (more than twice the size we used) and show better few-shot generalization. Works like [191] have shown the importance of having the right in-context examples for downstream performance leading to works that learn to retrieve relevant in-context examples [192].

Non-symbolic methods Most non-symbolic methods are sequence-to-sequence models trained on a large amount of question answering data [74, 95].

Symbolic methods Neural module networks like approaches parse complex questions into a pre-specified grammar and learn neural components to handle symbolic mathematical operations [178, 179, 172] which are recursively executed. State-of-the-art models on DROP, however, use a combination of BERT-based contextual models along with a calculator that performs discrete operations [180, 176, 193]. Works like Text Modular networks [174] and MRKL [173] are closest to our work. However, they are limited in the terms of types of simple questions they can answer (single-span only) and the complexity of reasoning they can do (single-order only). TMNs, additionally, use a classifier that scores the generated chains module and filters out incorrect question decompositions, while we use contrastive estimation to learn a better question decomposer and as a result do not need a chain scorer.

5.6 Conclusion

We present a way to successively decompose complex questions into simple QA pairs, which allows for modular QD and QA systems that can be trained and queried independently. When performing in-context learning, we showed that successive prompting yields an improvement of 4.6 F1 over chain-of-thought prompting. When replacing just the in-context QA module with a fine-tuned one, which is adept at handling list type questions, we further improve the overall performance by 9.5 F1. We believe that modular systems that decompose and delegate tasks to the most appropriate model, whether that is a large LM or a tailored component, are more effective at solving complex tasks than trying to have a large LM solve the entire task on its own. Successive prompting shows one way this decomposition and delegation can be done.

Part III

Generalizability: Challenges and Interventions

Chapter 6

Effectiveness of data augmentation for generalization

A mathematician who can only generalize is like a monkey who can only climb up a tree, and a mathematician who can only specialize is like a monkey who can only climb down a tree. In fact, neither the up monkey nor the down monkey is a viable creature.

George Pólya

(Father of Problem Solving)

Recent advances in open-domain question answering (ODQA) have demonstrated impressive accuracy on general-purpose domains like Wikipedia. While some work has been investigating how well ODQA models perform when tested for out-of-domain (OOD) generalization, these studies have been conducted only under conservative shifts in data distribution and typically focus on a

single component (i.e., retriever or reader) rather than an end-to-end system. This work proposes a more realistic end-to-end domain shift evaluation setting covering five diverse domains. We not only find that end-to-end models fail to generalize but that high retrieval scores often still yield poor answer prediction accuracy. To address these failures, we investigate several interventions, in the form of data augmentations, for improving model adaption. Furthermore, we leverage our evaluation set to establish a perspicuous relationship between the effectiveness of different intervention schemes and the specific types of dataset shifts considered. We propose a generalizability test that estimates the type of shift in a target dataset without training a model in the target domain and that the type of shift is predictive of which data augmentation schemes will be effective for domain adaption. Overall, our study emphasizes the importance of both generalization and specialization in ODQA and with the help of these interventions, we observe an increase in end-to-end performance by up to ~24 points.

6.1 Introduction

General-purpose open-domain question answering (ODQA; [194, 195, 196]) is an important task that automates reading and understanding a large corpus of documents to answer a given question succinctly. It is especially crucial in fields such as biomedicine, legal, news, etc., where more documents are added daily, outpacing the speed at which a user can process the information.

Current state-of-the-art ODQA systems perform a two-stage pipeline process [196]: 1) Given a question and document corpus, a *retriever* [150, 197, 33] selects relevant passages and 2) a question answering model, also known as a *reader* [198] answers the given question based on the retrieved passages. This decoupling allows for independent advances in domain adaptation of general-purpose retrievers [199] and question-answering [200] models.

To enable practical application, an ODQA system should assist humans in keeping up with new

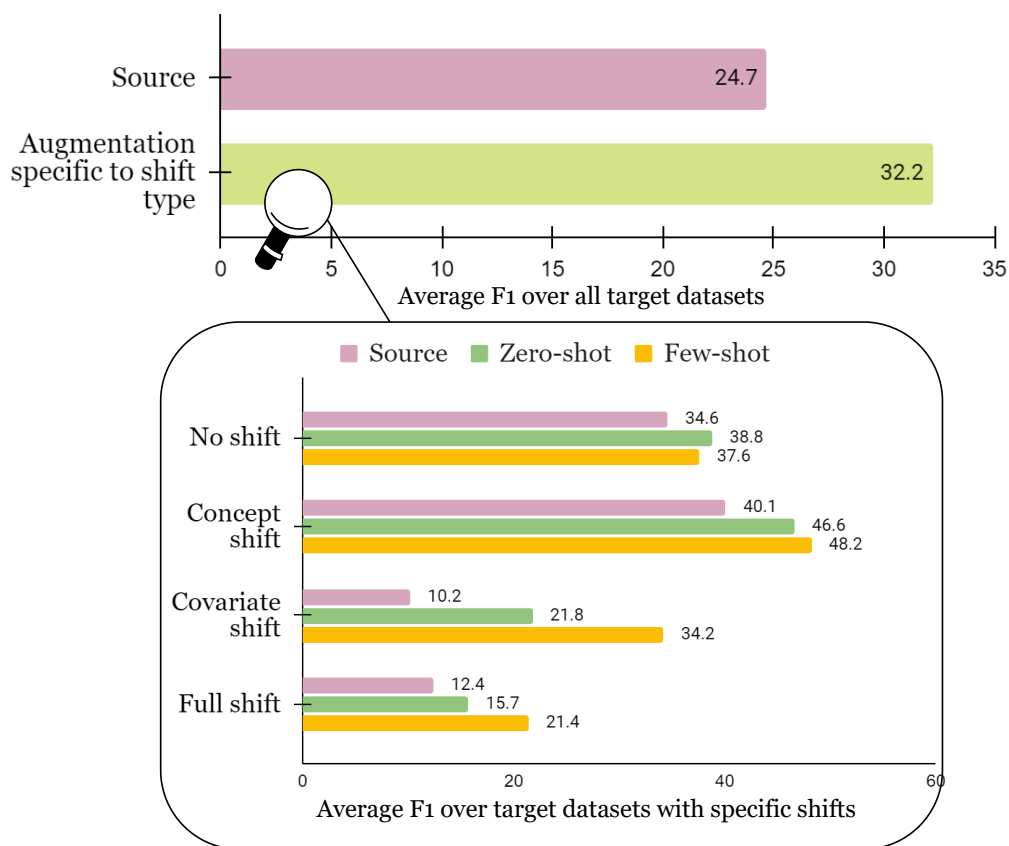


Figure 6.1: **Effect of interventions on dataset shifts.** *Top:* Average end-to-end performance of source domain model is quite poor when applied to OOD datasets. Source model (trained on general-purpose domain) performance improves when adapted to unseen target domain with interventions. *Bottom:* Drill-down of performance into zero and few-shot data augmentations averaged over target datasets exhibiting these shifts shows covariate and concept shifts respond to zero and few-shot data augmentations. Target datasets with No shift do not improve much with any intervention while full shift benefits most from Few-shot.

knowledge without requiring annotations for every new domain or concept. For this, the system should be resilient to changes in the document, question, and answer distributions. Unfortunately, the current work in ODQA focus solely on Wikipedia corpus and do not study effectiveness of a model trained on such a general-purpose domain when applied to an unseen domain. To gauge how likely it is for a source domain model to succeed on an unseen domain we need to understand its ability to work out-of-the-box or even adapt to a new target domain, under varying types and degrees of dataset shifts. [38].

In this work, we study the challenges and interventions for generalizing ODQA models to new domains via four contributions. First, to understand how well state-of-the-art ODQA system (trained on general-purpose domain) performs on a variety of target distributions, we define a collection of datasets for evaluating domain generalization. We aggregate a set of seven ODQA datasets spanning five different domains (§6.2). We observe that source ODQA model does not generalize well (Figure 6.1, Top) on this collection (§6.4). Second, to automatically determine the type of data shift with only a small number of labeled target domain examples we propose a *generalizability test*. This test assesses the type and degree of shift, a new domain suffers with respect to source domain (§6.3). Third, to understand the adaptability of the source model to a target domain, we analyze the performance of various intervention schemes, including existing zero-shot in-domain question generation and a novel few-shot language model-aided generation. These schemes create data akin to target domain which is augmented with source domain to learn an adapted version of the source model. Overall, we observe improvement in performance across all the target datasets (Figure 6.1). The degree of improvement depends on the intervention scheme and underlying dataset shift (§6.5). Finally, we propose a simple and effective few-shot method which improves the performance by up to 24% in F1. This method prompts a large language model with 8 examples to generate examples for further adaptation.

Putting it altogether, we use the generalizability test to gauge the type and degree of dataset shift in a target dataset. Then, we empirically show that certain types of dataset shifts respond well to specific intervention schemes (§6.5, Figure 6.1). This helps ascertain whether we can adapt a source model to unseen domain with minimal supervision.

6.2 Background and Evaluation Setup

An ODQA model learns interactions among three random variables: Question (\mathbb{Q}), answer (\mathbb{A}) and context (\mathbb{C}). For a given $q \in \mathbb{Q}$, first the retriever \mathcal{R} returns a set of passages, $C_q = \mathcal{R}(q, \mathbb{C})$. These

passages are then sent to a reader model \mathcal{M} to obtain the final answer, $\hat{a} \leftarrow \mathcal{M}(a|q, C_q)$.

Following prior work, we evaluate retriever performance with the Acc@k metric, which computes if the oracle answer is found in the top- k retrieved passages¹. We set $k=100$ in all of our experiments. For reader performance, we compute token-level F1 between the oracle and predicted answer.

6.2.1 Datasets

We test the generalization capabilities of a model trained on a *source domain* when applied to seven datasets in five very different *target domains*.

Source Domain: For source domain we use documents from English Wikipedia and QA pairs for supervision from NaturalQuestions (NQ) [148] and BoolQ [77]. We treat this domain as our source as it is used for the vast majority of current work in ODQA (and many other areas of language research). In addition to the supervised training data from NQ and BoolQ, we also consider cloze-style questions derived from the QA pairs in NQ. For each QA pair, we retrieve a sentence from Wikipedia with the highest BM25 similarity score. We convert the retrieved sentence into a cloze-style question by replacing the answer string in the sentence with sentinel markers [33]².

Target Domains: For our target corpora, we re-purpose seven open-domain QA and/or reading comprehension datasets spanning five different domains (Stack Overflow, Reddit, Pubmed, Japanese Statute Law codes, CNN/DailyMail, and Wikipedia). The datasets Quasar-S [201], Quasar-T [201], SearchQA [146] and BioASQ [202] were introduced as ODQA datasets over Stackoverflow, Reddit, Wikipedia, and Pubmed corpus respectively. Additionally, we re-purpose reading comprehension datasets, NewsQA [203] and CliCR [204] as ODQA datasets, by retrieving a set of passages for

¹The only exception is COLIEE dataset which primarily contains boolean (yes/no) answers so we instead use oracle passage to compute Acc@100.

²We use cloze augmentation for training reader models because some target datasets contain cloze-style questions, keeping the question distribution consistent across different experimental setups. We do not perform this augmentation for retrievers because we observed a performance drop in initial experiments.

each QA pair from Pubmed and CNN/Dailymail corpus. For COLIEE [205], we convert the original entailment questions into boolean questions and retrieve passages from legal code statutes provided with the task. We confirm that these reading comprehension datasets can be reasonably re-purposed for our ODQA setup by achieving a reasonable end-to-end performance of ODQA models trained on gold target domain QA pairs with BM25 retrievals from the target corpus (UB-Ret, Figure 6.3).

6.2.2 Models

We compare four **retrievers**: (1) BM25 [206] (sparse and unsupervised), (2) Contriever, semi-supervised with MS-MARCO [197], (3) Dense Passage Retriever (DPR) [150], and (4) the state-of-the-art source domain model Spider [207]. DPR and Spider are dense and supervised.

As for **reader**, we use the state-of-art fusion-in-decoder (FiD) model [198] that uses the top 100 documents to generate the final answer.

6.3 Generalizability Test

There are many aspects that determine in what ways and to what extent one data distribution differs from another. It is often challenging to quantify the degree of *generalizability* or diverseness for a new domain without collecting enough samples to train a model in the new domain. To address this issue, we propose a method to assess the type and degree of diversity by utilizing only a few examples from the target domain as an evaluation set.

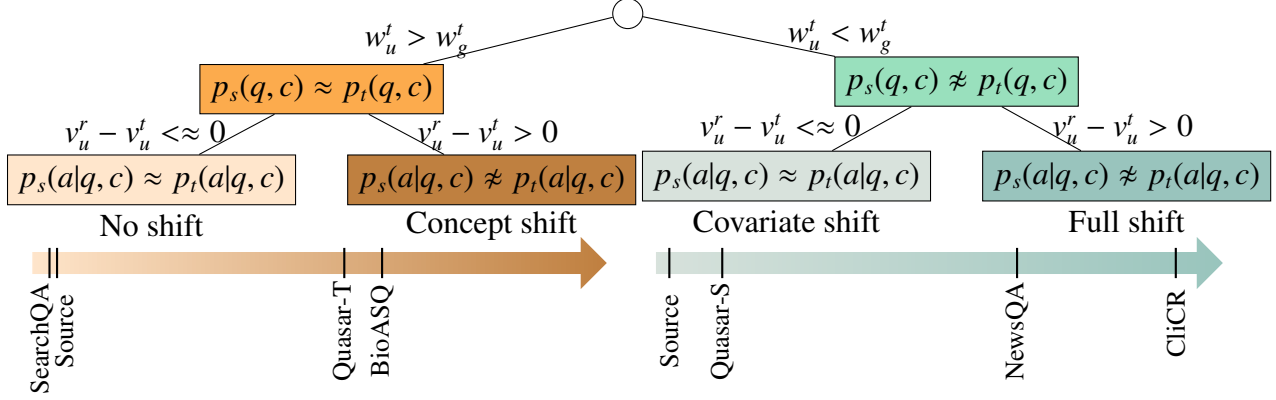


Figure 6.2: **Generalizability Test:** At the first level, we decide whether the input distribution is closer to the uniform distribution or gold. At the second level, the gradual increase from left to right in the leaf nodes depicts decrease in distance of output distribution from uniform. The target datasets at the bottom are placed based on distances in Table 6.1. The nodes represent if the source model $p_s(a|q, c)$ is compatible or not with the target dataset $p_t(a|q, c)$

6.3.1 Types of dataset shift

Different types of dataset shifts [38] have been proposed in the literature but they are often studied in a classification setup. For our application, we consider *concept* and *covariate* shifts which are more amenable to our pipelined ODQA setup — with input as a joint distribution over question and contexts and output as a distribution over answers given question and contexts as input.

No shift occurs when the input and output distributions match across the source and target domains.

Concept shift [208] occurs when the input distribution of the source and target domains match, i.e., $p_s(x) = p_t(x)$ while the output distribution between source and target domain does not match, $p_s(y|x) \neq p_t(y|x)$.

Covariate shift [209] occurs when the source and target input distributions do not match, i.e. $p_s(x) \neq p_t(x)$ while the output distributions match $p_s(y|x) = p_t(y|x)$.

Full shift occurs when both the source and target input and output distributions do not match.

6.3.2 Calculating shift for ODQA

We characterize the shift in ODQA as a two-step process. First, we compute the input distribution, i.e, the joint question and context distribution using un-normalized (energy) scores from a dense retriever [150] that quantifies the compatibility between a given question and a context via $\mathcal{R}(q, c)$. Then, we normalize the scores from the retriever over a set of contexts. Ideally, the set of contexts should be the entire target domain document corpus, however, that can be prohibitively computationally expensive and also results in a high entropy distribution. Instead, we use a subset of contexts, C , from the entire corpus \mathbb{C} . We ignore the prior over questions since it remains constant when calculating the context distribution for a specific question. Instead, we approximate the joint with conditional distribution over contexts given question.

$$p(q, c) \propto \frac{\mathcal{R}(q, c)}{\sum_{c_k \in C} \mathcal{R}(q, c_k)} \quad (6.1)$$

In the second step, we test whether the output distributions match by computing the likelihood of generating the oracle answer given a question, q , and the relevant contexts, C_q . In an ideal scenario, we can do this by performing global normalization [115] over all possible answer spans in the corpus which is intractable. Instead, we use a sub-sample of answers, \mathcal{A} , to compute the output distribution as shown below.

$$p(a|q, C_q) = \frac{\prod_t \mathcal{M}(a^t|a^{\leq t}, q, C_q)}{\sum_{a_k \in \mathcal{A}} \prod_t \mathcal{M}(a_k^t|a_k^{\leq t}, q, C_q)} \quad (6.2)$$

Dataset	Retriever ($w_u^t - w_u^s$)	Reader ($v_u^r - v_u^t$)	Shift
BioASQ	0.30	0.17	Concept
CliCR	-0.88	0.23	Full
Quasar-S	-0.66	0.07	Covariate
Quasar-T	0.20	0.16	Concept
NewsQA	-0.19	0.18	Full
SearchQA	0.61	0.00	No

Table 6.1: **Wasserstein distance** computed over 100 labeled examples from the target set. The negative retriever value implies that the target dataset falls on the right side of decision tree at first level (Figure 6.2).

6.3.3 Predicting type of dataset shift

To compute the type of shift (§6.3.1), we need a model trained on the target domain (p_t) which requires a large number of examples. However, our goal is to determine if a source model can be adapted to the target dataset with only a few examples for target evaluation. To do this, we conceptualize adapting or fine-tuning a pre-trained source model as a Bayesian framework. In this framework, the source model acts as a prior which when exposed to interventional data (for adapting) and target data (for fine-tuning), results in an adapted or fine-tuned posterior distribution. If the prior (source model) contains an informative signal with respect to the target dataset then we do not require much supervision to learn an effective posterior. However, if the prior is non-informative we need a lot of supervision to learn the posterior.

Towards this end, we devise a *generalizability test*, where we use a small set of evaluation examples sampled from each target dataset to compute input and output distribution using the source domain model. Then, we compare these distribution with the a non-informative prior like uniform distribution and informative prior like the oracle distribution to gauge if the source model is closer to uniform or oracle distribution. This helps us assess the effectiveness of the source model towards the target dataset without having to train a model in the target domain.

Input/Retriever Distribution: To determine if the input distribution contains informative signal with respect to target evaluation set, we need to compute the distance of the input distribution from uniform and oracle distribution. To do this, we follow Eq. 6.1 and compute the input distribution, with passages from across examples in the entire target evaluation set as the subset for normalizer computation. Then, for a each question, we compute the Wasserstein distance, w_u^t , [210] between the input distribution and the uniform distribution and average these values over all the examples in the target evaluation set. Similarly, we also compute the distance between the gold or oracle distribution and the input distribution as w_g^t . If $w_u^t > w_g^t$, we conclude that the target distribution is far from the uniform distribution and closer to the gold distribution, indicating that the source model is compatible with the target distribution (Figure 6.2).

Output/Reader Distribution: In similar vein as input distribution, we need to compare the output distribution with corresponding uniform and oracle distribution over answers. To do this, we follow Eq. 6.2 and compute the output distribution, with set of answer spans from across all the examples in the target evaluation set for normalizer computation. Then, we compute the Wasserstein distance between the uniform and output distribution averaged over the target evaluation set as v_u^t .

In an ideal scenario, we would compare the distance between and oracle and output distribution with v_u^t , similar to input distribution. However, empirically we find that output distribution is always closer to uniform than oracle, even when evaluated with source distribution. We believe this is because of two reasons. First, the conditional answer generation model (\mathcal{M}) is not trained with a contrastive loss like the retriever, resulting in a high entropy answer likelihood distribution. Second, the support set of answers used for normalization contains only grammatically correct answer spans making the likelihood scores attenuated. To address these issues, we use a reference answer conditional distribution to de-bias the likelihood scores with a threshold. To obtain this threshold, we consider the source distribution as a reference and compute the distance between output distribution evaluated on examples from source evaluation set and the uniform distribution

as v_u^r . Since the reference based output distribution is in-domain, it should be far from the uniform distribution and closer to oracle distribution. As a result, if $v_u^r - v_u^t$ is close to 0, we assess that the target is far from uniform and that source model is compatible with the target dataset.

In Figure 6.2, we put this altogether as a decision tree to identify the type of dataset shift. We observe that SearchQA falls under the *No shift* category as it is close to the source domain, hence, we conjecture that it will observe minimal improvements under most data intervention schemes as the source model already captures the target distribution (§6.5). We also conjecture that datasets falling under *Concept shift* and *Covariate shift* are more amenable to zero-shot data interventions, while, *Full shift* would benefit more from few-shot or in-domain annotations from the target domain. We consider few shot augmentations as a proxy for annotating examples in the target domain because they are generated with supervision from target dataset.

6.4 How Well do Models Generalize?

We test the OOD performance of the source model on target datasets and analyze the failures.

6.4.1 Reader Generalization

In Figure 6.3, we test the end-to-end performance of three model variants:

Source: a reader trained with source dataset and contexts retrieved by BM25, demonstrating zero-shot generalization performance.

Upperbound-Reader (UB-READ): a reader trained on the target dataset with contexts retrieved by BM25 – the overall strongest retriever.

Upperbound-Retriever (UB-RET): a reader trained on the target dataset with gold contexts to

approximate upper-bound performance.

We observe large performance drops when evaluating the source model on target domains (Figure 6.3), especially when the target corpus differs from Wikipedia, such as in Quasar-S (Stack Overflow) and CliCR (PubMed), even though the model requires similar reading capabilities to those needed in the source domain. Interestingly, even though BM25 retriever accuracy is relatively high on the target datasets (Figure 6.4, $\sim 83\%$ Acc@100 on Quasar-S), that accuracy does not translate to strong reader performance (Figure 6.3, $\sim 11\%$ F1 on Quasar-S).

To understand this performance gap, we manually sample 50 predictions from each target dataset where retrieved passages contain the oracle answer but the reader produced an incorrect prediction. We observe that in **$\sim 65\%$ cases, the Acc@100 metric yields a false positive**, where the passage contains an exact string match of the correct answer, but the context does not actually answer the given question. In other cases, the reader is unable to understand the context. For example, for the question: `What is the name of the office used by the president in the white house?` and answer: `oval`, the retrieved passage: `A tunnel was dug into the White House connecting the Oval Office to a location in the East Wing....` is credited (incorrectly) as context answering the question.

6.4.2 Retriever Generalization

We compare the zero-shot generalization of four retrieval models in Figure 6.4. Spider, which is the best performing model on the source domain, exhibits improvement on SearchQA ($\sim 1\%$) (which is similar to source distribution), but shows large drops in performance when applied to the target datasets: $\sim 40\%$ on NewsQA, $\sim 28\%$ on Quasar-T and, Quasar-S. To understand the drop, we manually analyze 50 random incorrect predictions from Spider. We observe two major failure modes. First, we find that dense models are sensitive to changes in the length of contexts. When exposed to documents with heterogeneous lengths, models tend to over-retrieve shorter contexts. To

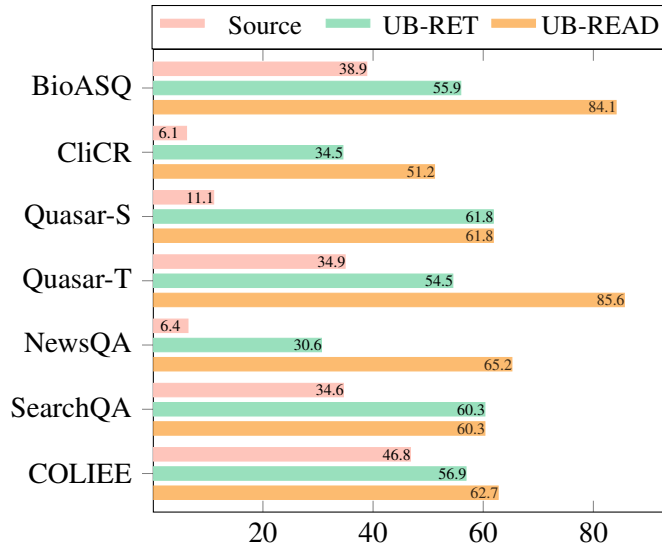


Figure 6.3: Reader performance on the target set without any interventions. SearchQA, Quasar-S and Quasar-T do not have gold passage annotations and so UB-READ does not improve over UB-RET. The majority voting baseline on COLIEE is 50.95.

quantify the sensitivity to changes in lengths on source domains itself, we pool passages from all target corpus into a combined index. We observe that the performance of Spider when exposed to this combined index reduces by $\sim 15\%$ and restricting the minimum length of contexts to 50 words alleviates the problem and recovers the original performance. The second common failure mode occurs due to changes in distribution of entity types from source to target. For example, words like `plant` in `Which is produced in plants of narora kakrapar tarapur` refers to `power plant` in Wikipedia, while in case of PubMed it often refers to living organic matter [211]. Overall, BM25, being an unsupervised method, has the best performance across all domains.

6.5 Interventions for Improving Adaptation

Domain Adaptation is shown to be a causal intervention [212] mechanism to effectively understand impact of an augmentation technique without much concern about spurious correlations.

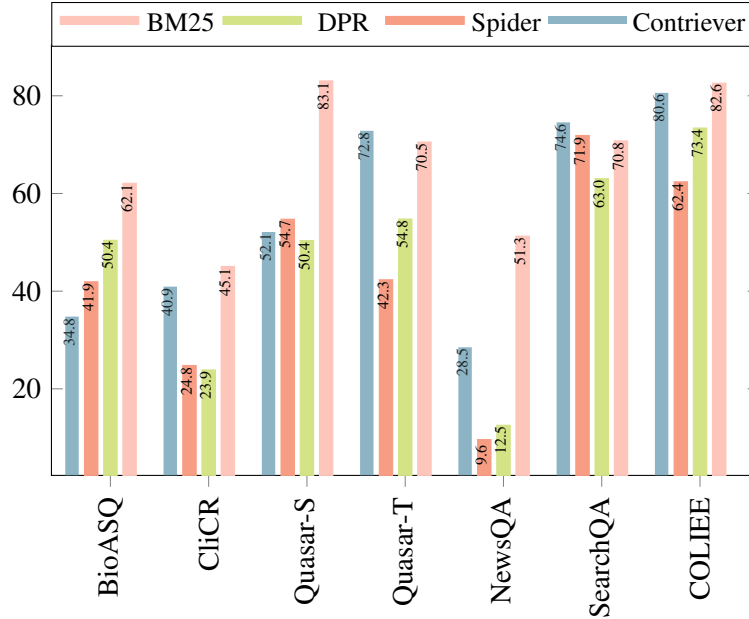


Figure 6.4: Retriever performance (Acc@100) without any interventions on target domain corpus

6.5.1 Zero-shot adaptation methods

We perform a set of zero-shot data intervention methods, where we consider the effect of change in distribution of each random variable: Question, answer and context one at a time, while keeping the other two fixed.

Varying context distribution To test the effect of change in context distribution, we pool passages from all corpora into a combined index. We observe that supervised models like Spider are sensitive to out-of-domain distractors, unlike BM25, especially when the target dataset uses same corpus as source (Wikipedia). For example, SearchQA suffers a performance drop of $\sim 15\%$. On average we see a performance improvement of $\sim 2\%$ (w/o COLIEE) when the target index is changed to the combined index. BM25 still out-performs Spider on average by 19.1% with the combined index. However, we observe a drop in performance of the FiD reader of up to $\sim 5\%$ in F1 for NewsQA with the combined index. More details are in the appendix (Figure 6.5 and 6.6.)

Augmentations	Retriever	Reader
Random	45.35	33.50
Uniform	50.02	39.07
Most frequent	39.33	38.18
BioASQ train answers	47.48	41.33

Table 6.2: Answer distribution: Retriever (DPR) and Reader (FiD with BM25 retrievals) F1 on BioASQ.

Varying answer distribution Many works [139, 13, 2] have shown that bias in the answer prior distribution can introduce spurious correlations in model learning. This effectively improves the model performance at the cost of OOD generation. To test whether we can improve the performance of adapted source model by varying the answer distribution, we experiment with a variety of answer distributions over plausible set of answer spans. To obtain the set of answer spans, we extract and annotate coarse-grained entity types from the target corpus using spaCy³. We use this coarse-grained entity type information as a set of classes from which to choose 50k entities with four different sampling strategies: Most frequent, uniform, randomly sampled based on entity type categories, and sampling in proportion to entity type distribution of answers in the target training set.

The source model has reasonable end-to-end performance on BioASQ, even with passages from the source corpus (Wikipedia), suggesting that it contains sufficient information for answering many BioASQ questions. Consequently, we select BioASQ for these controlled experiments (Appendix Figure 6.6). This allows us to use the Wikipedia corpus alone for retrieval, which makes it easier to fix the passage distribution. In Table 6.2, we show that uniform sampling boosts retriever performance compared to random sampling, allowing the model to learn from all types of answers and generalize better to unseen answer distributions. On the other hand, the best reader model performance is when we know the correct answer distribution of the target dataset up front, showing that the answer priors influence reader performance. However, in a zero-shot setup, we do not have access to this distribution, so we adopt the second-best technique, uniform sampling from across the entity type categories, in the following experiments.

³<https://spacy.io/>

Dataset	Retriever			Reader		
	Source	ClozeQA	QGen	Source	ClozeQA	QGen
BioASQ	50.41	48.0	45.4	45.3	49.4	46.4
CliCR	23.8	24.9	23.9	6.12	7.34	10.5
Quasar-S	50.3	66.8	68.2	10.2	21.7	17.4
Quasar-T	54.7	53.9	55.5	34.9	41.9	44.7
NewsQA	12.5	18.7	15.2	18.5	21.2	12.7
SearchQA	63.0	52.9	54.7	34.6	38.8	37.2
COLIEE	61.4	60.5	57.8	46.7	54.1	62.3

Table 6.3: Zero-shot: Comparing retriever (DPR) and reader (FiD with BM25 retrievals) performance on two types of question formats for augmentation.

Varying question distribution We vary the question distribution by augmenting the source domain with QA pairs generated using two different methods. Our first approach (QGen) uses a question generation model [213] trained on the source domain to generate a question given a passage and an answer. This question generation model is applied to a new target passage and a plausible answer span from the passage [214, 215, 216, 217]. The second approach (Cloze QA), which has been less explored previously, converts a sentence in the target corpus to a fill-in-the-blank style cloze question [72] by masking a plausible answer span (entity mention) in the sentence. We sample answer spans uniformly based on an entity type distribution from the target corpus and then query our combined index to create a dataset containing cloze style questions aligned with relevant documents. We use these same sampled answers to generate standard QGen QA pairs as well. We combine these data interventions with our initial source domain data to train a DPR retriever and a FiD reader (Table 6.3). We observe similar average performance across both intervention types in retriever and reader models. However, cloze QA pairs are computationally much more efficient to generate as they do not require additional question generation models.

Discussion on generalizability test In §6.3, we hypothesized that datasets with less severe shift (Quasar-S, Quasar-T, and BioASQ) would show more performance improvements with zero-shot adaptation as compared to datasets with severe shift (CliCR and NewsQA). Indeed, we observe an avg. improvement of about 8.5% F1 on datasets having Concept and Covariate shift while only

3.5% F1 on datasets with Full shift in Table 6.3. Moreover, in Figure 6.1, we see that target datasets with *No shift*, do not show much improvement with any intervention as the source model already captures the distribution. Datasets with *Full shift* need few-shot examples for better adaptation while datasets with *Concept* and *Covariate* shift are able to adapt with zero-shot data interventions.

6.5.2 Few-shot Generalizability and Adapatability

Zero-shot adaptation does not work well when the target distribution is far from the source. For these cases, we experiment with few-shot adaptation.

Few-shot data generation Zero-shot interventions like QGen are trained on the source and do not produce generations that are fully compatible with the target domain and thereby do not provide much useful signal. An alternative approach would be to train a question generation model with a few examples from the target domain. However, it is difficult to adapt or fine-tune a question generation and answering model (for validating QA pair correctness) with very few examples.

Dataset	Retriever		Reader		Closed Book (F1)
	Baseline	DataGen	Baseline	DataGen	
BioASQ	50.4	51.3	45.3	50.6	32.0
CliCR	23.8	29.0	6.12	19.4	10.8
Quasar-S	50.3	71.9	10.2	34.2	23.7
Quasar-T	54.7	55.4	34.9	45.8	55.3
NewsQA	12.5	22.7	18.5	23.3	8.67
SearchQA	63.0	63.3	34.6	37.6	61.5
COLIEE	73.3	82.2	46.8	61.1	53.0

Table 6.4: Both Closed Book and DataGen use eight few-shot examples from the target domain. Closed Book LLM contains 540B params while the Retriever and Reader contain 110M and 770M params respectively. Closed-book performance for NQ is 36.71.

To capture target distribution without a lot of supervision, we propose a few-shot technique (DataGen) that prompts a large language model (LLM; [96]) to generate a sentence given a passage.

We use eight seed examples from the target domain to generate additional training data to help bootstrap adaptation in the target domain. We observe that it is easier for large language models to condition on a single variable (context) and compress [218] multiple facts from the passage into a single sentence, as compared to conditioning on a context and answer span together. Moreover, in §6.5.1 we observed that augmentation with cloze-style QA pairs yields similar performance to using question-formatted QA pairs, offering evidence that the precise format is not as important as the content itself.

We prompt the model in the following format: `After reading the article, <<context>> the doctor said <<sentence>>` for PubMed articles. For other target corpora we replace `doctor` with `engineer`, `journalist`, and `poster` for Stack Overflow, DailyMail, and Reddit respectively. To filter out invalid sentences, we remove any generation that: 1) includes a number, 2) does not repeat part of the passage verbatim, and 3) has less than 75% word set overlap with the passage (after removing stopwords). To gauge the precision of our generations, we sampled 20 generated sentences for each dataset and found that they are correct more than 70% of the time. To test retriever performance, we train a DPR model with source domain data and ~8k examples containing pairs of original passage and generated sentence for each target dataset. We observe performance improvements of ~18% on NewsQA, ~13% on CliCR, and ~24% on Quasar-S (Table 6.4). Moreover, LLMs contain substantial factual knowledge in their parameters and we observe that they do particularly well in a closed-book setting on datasets with trivia-based factual questions, like SearchQA and Quasar-T, but do not perform well in other cases. Following our conjecture in §6.3, datasets with *Full shift* on average show an improvement of 12.1% with few-shot interventions, compared to 3.5% with zero-shot, which is also evident in Figure 6.1. We show qualitative examples in Appendix (Figure 6.7).

6.6 Related Work

Domain generalization in readers The most popular work in generalization in reading comprehension was introduced as part of the MRQA [200] challenge, which focuses on transfer learning from multiple source datasets to unseen target datasets. This multi-task learning setup requires the model to perform complex reasoning at test time that may be unseen at training. However, in this work, we focus on generalization capabilities of an end-to-end ODQA setup that is able understand passages in the new domain, and not the abilities to perform unseen reasoning.

Domain generalization in retrievers A recent line of work that tests domain generalization of retrievers [219, 207, 196] focuses on conservative changes to source domain, for instance testing generalization of model trained on Natural Questions applied to WebQuestions [220] or TriviaQA [147], all of which use the same Wikipedia corpus. BEIR is a recent retrieval benchmark, [199] tests generalizability of only the retriever in isolation and not end-to-end ODQA performance which is a brittle metric.

Domain adaptation work in retrievers [221] generate passages using few shots but do not require the answer to be correct. [222] performs a zero-shot adaptation using noisy labels for retrievers. [223] utilizes examples from target domain in a transfer learning setup while we work in a zero to few shot setting.

Domain generalization in other tasks Incidental supervision signals in [224] determine which dataset has a useful signal for a target classification task. Similar to [200], in machine translation various works [225, 226] learn to balance positive and negative feature transfer from multiple source domains to a target domain.

6.7 Appendix

We used JAX on TPUs for training reader models and PyTorch on GPU for training retriever models. We used open-source github implementations for DPR ⁴, Contriever ⁵ and Spider ⁶. For retrieving top-100 passages for reader input, we used ScaNN⁷ library. We use T5-base model for reader and BERT-base for retriever. We fine-tune the retriever and reader with learning rate 1e-5 and 5e-5 respectively.

6.7.1 How are evaluation sets curated?

We consider validation sets from each of the target dataset, BioASQ, CliCR, Quasar-S, Quasar-T, NewsQA, SearchQA, COLIEE as part of our evaluation set. SearchQA, Quasar-S and Quasar-T were already published as ODQA datasets so we used them as it is while we had re-purpose some of the other datasets that were not originally ODQA dataset by processing them as described below.

COLIEE: The COLIEE Shared Task 4⁸ provides a list of Japanese legal codes in English language. To convert these legal codes from a flat list into paragraphs, first we split them into specific article sections with regex string "Article [0-9]+ ". We further split each article into passages containing a maximum of 256 words.

NewsQA: We created an index on CNN/Dailymail documents by splitting them into passages of 256 words and pooled them together to create a corpus.

⁴<https://github.com/facebookresearch/DPR.git>

⁵<https://github.com/facebookresearch/contriever.git>

⁶<https://github.com/oriram/spider>

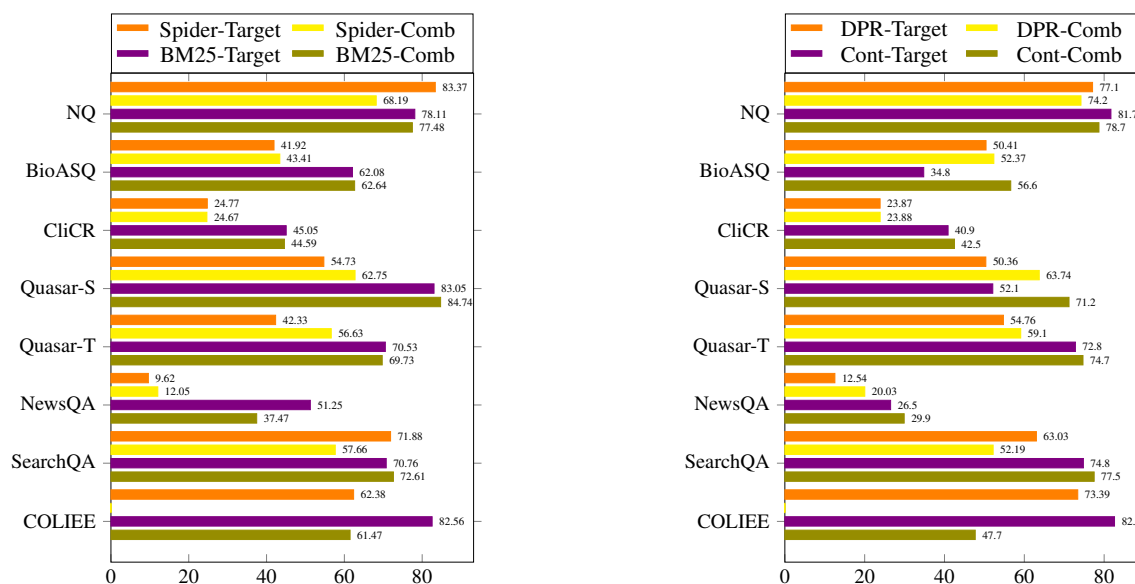
⁷<https://ai.googleblog.com/2020/07/announcing-scann-efficient-vector.html>

⁸<https://sites.ualberta.ca/~rabelo/COLIEE2022/>

CliCR and BioASQ: We used PubMed corpus published as part of BEIR [199] benchmark. We split the pubmed abstracts in this corpus into passages of size 256 words.

6.7.2 Varying context distribution

As described in §6.5.1, we test retriever (Figure 6.5) and reader performance (Figure 6.6) when exposed to different set of passage. Figure 6.6 shows reader performance with passages retrieved with BM25 on source (i.e. wikipedia), target (i.e. respective target corpus) and combined (i.e. all corpora pooled together). Figure 6.5 compared performance of Spider and BM25 with Target (i.e. dataset specific target corpus) and Comb (i.e. all corpora pooled together)



(a) Spider-Comb has 0% Acc@k on this COLIEE due to a large number of distractors.

(b) DPR-Comb has 0% Acc@k on this dataset due to a large number of distractors.

Figure 6.5: Retriever Performance (Acc@100): Varying context distribution by creating a combined document index. For COLIEE, we use oracle passages for performance computation.

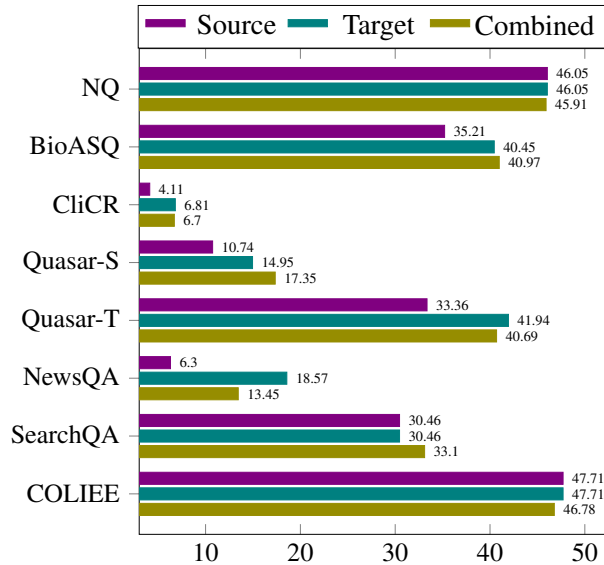


Figure 6.6: Reader Performance (F1): Effect of change in context distribution with BM25 retrievals from the combined index.

6.7.3 Varying answer distribution and pre-training corpus

Following §6.5.1 we try to understand the impact of pre-training and fine-tuning corpus on answer distribution. We do this by comparing the performance of the FiD reader initialized from T5 pre-trained on common-crawl dataset(C4) compared to one that was pre-trained on PubMed articles (Table 6.5). After pre-training, both models are then fine-tuned on our source domain data. In this case, we observe that fine-tuning on a domain that differs from that used in pre-training results in deterioration of model performance.

Augmentations	C4	Pubmed
Random	33.50	33.51
Uniform	39.07	35.97
Most frequent	38.18	34.90
BioASQ train answers	41.33	36.71

Table 6.5: Answer distribution: Reader performance on BioASQ with C4 and Pubmed pre-trained T5

6.7.4 Degree of domain shift

In Table 6.1, we showed only differences that governed which side of decision tree the shift types were categorized into, while here we show all the raw distance values.

Dataset	Retriever		Reader	
	w_u^t	w_g^t	v_u^t	$v_u^r - v_u^t$
BioASQ	0.6477	0.3450	0.1160	0.1765
CliCR	0.0602	0.9448	0.0573	0.2352
Quasar-S	0.1658	0.8355	0.2158	0.0767
Quasar-T	0.5978	0.3962	0.1231	0.1694
NewsQA	0.3992	0.5959	0.1125	0.1800
SearchQA	0.80350	0.1870	0.2988	-0.0063

Table 6.6: Wasserstein distance computed over 100 target domain examples. The distance between reference (source) and uniform over 100 validation set source domain examples is $v_u^r=0.2925$

6.7.5 Statistical Significance

The number of examples in all datasets except COLIEE are in the order of thousands, making the performance improvements significant. In the case of COLIEE, which has a boolean output space (i.e. answers are yes/no), we performed a binomial test to test the significance of few-shot reader performance in Table 6.4. The number of samples $n = 116$ (number of test examples), $p_0=0.468$ and $p_t=0.616$. We will reject the null hypothesis that baseline and few-shot distribution are equivalent, when $P(X \geq p_t * n) \leq 0.05$, where X is drawn from a binomial distribution, i.e., $X \sim B(n, p_0)$ [227] and we can compute the L.H.S to be, $P(X \geq 0.616 * 116) = 0.00006$ making it significant.

6.7.6 Qualitative examples generated by the few-shot method

Dataset, Corpus	Passage	Generated Sentence
BioASQ, Pubmed	Herceptin is widely used in treating Her2-overexpressing breast cancer. However, the application of Herceptin in prostate cancer is still controversial.... This implies that targeting Her2 by both radio- and immunotherapy might be a potential strategy for treating patients with androgen-independent prostate cancer...	Herceptin is a breast cancer drug that has been used in treating prostate cancer.
CliCR, Pubmed	An infant was admitted with symptoms of diarrhoea and vomiting. After initial improvement she unexpectedly died. Postmortem confirmed a diagnosis of cytomegalovirus (CMV) enterocolitis. The authors report this case and review other published cases of immunocompetent infants who presented with this infection. Clinicians should consider stool CMV PCR test or referral for endoscopy and biopsy in young babies who present with profuse and prolonged episodes of diarrhoea.	Immunocompetent infants can present with CMV enterocolitis.
Quasar-S, Stack-overflow	I've recently found scala-bindgen from a Gitter room on Scala Native. Seems like at the present point in time they are developing a tool for generating Scala bindings for C header-files. Are there plans for generating Scala bindings for Objective-C and C++ too...	scala-bindgen is a tool that generates scala bindings for C header files.
Quasar-T, Reddit	Interview With Gary James' Interview With Marshall Lytle of Bill Haley's Comets It can be safely said that "Rock Around The Clock" was the song by the group Bill Haley And His Comets that started the Rock 'n Roll movement. Still performing today, he spoke about those early days of Rock 'n Roll and his appreciation for what it meant to him.	Bill Haley and his comets made rock and roll music
NewsQA, CNN/Daily-mail	The Kardashians are already a staple on E! Network . But they've chosen the month of November to assert their dominance on the book world. Kourtney, Kim, and Khloe's first novel," Dollhouse ," hits shelves today . "Dollhouse," the first fiction endeavor from the Kardashians, follows sisters Kamille, Cassidy, ...	The Kardashians released a new book called 'Dollhouse'.
SearchQA, Wikipedia	Charles Henry Dow was an American journalist who co-founded Dow Jones and Company with Edward Jones and Charles Bergstresser. Dow also founded The Wall Street Journal, which has become one of the most respected financial publications in the world... In 1877, he published a History of Steam Navigation between New York and...	Charles Henry Dow, an American journalist, founded The Wall Street Journal in 1882.

Figure 6.7: Examples of data generated from few-shot prompting.

Chapter 7

ORB: Open-Reading Benchmark

”In God we trust, all others must bring
data.”

W. Edwards Deming

(father of the quality management)

Reading comprehension is one of the crucial tasks for furthering research in natural language understanding. A lot of diverse reading comprehension datasets have been introduced to study various phenomena in natural language, ranging from simple paraphrase matching and entity typing to entity tracking and understanding the implications of the context. Given the availability of many such datasets, comprehensive and reliable evaluation is tedious and time-consuming for researchers working on this problem. We present an evaluation server, **ORB**, that reports performance on seven diverse reading comprehension datasets, encouraging and facilitating testing a single model’s capability in understanding a wide variety of reading phenomena. The evaluation server places no restrictions on how models are trained, so it is a suitable test bed for exploring training paradigms and representation learning for general reading facility. We also collect and include synthetic augmentations for these datasets, testing how well models can handle out-of-domain questions.

7.1 Introduction

Research in reading comprehension, the task of answering questions about a given passage of text, has seen a huge surge of interest in recent years, with many large datasets introduced targeting various aspects of reading [76, 17, 116, 228]. However, as the number of datasets increases, evaluation on all of them becomes challenging, encouraging researchers to overfit to the biases of a single dataset. Recent research, including MultiQA [229] and the MRQA workshop shared task, aim to facilitate training and evaluating on several reading comprehension datasets at the same time. To further aid in this direction, we present an evaluation server that can test a single model across many different datasets, including on their hidden test sets in some cases. We focus on datasets where the core problem is natural language understanding, not information retrieval; models are given a single passage of text and a single question and are required to produce an answer.

As our goal is to provide a broad suite of questions that test a single model’s reading ability, we additionally provide synthetic augmentations to some of the datasets in our evaluation server. Several recent papers have proposed question transformations that result in out-of-distribution test examples, helping to judge the generalization capability of reading models [230, 231, 232]. We collect the best of these, add some of our own, and keep those that generate reasonable and challenging questions. We believe this strategy of evaluating on many datasets, including distribution-shifted synthetic examples, will lead the field towards more robust and comprehensive reading comprehension models.

Code for the evaluation server, including a script to run it on the dev sets of these datasets and a leaderboard showing results on their hidden tests, can be found at <https://leaderboard.allenai.org/orb>

7.2 Datasets

We selected seven existing datasets that target various complex linguistic phenomena such as coreference resolution, entity and event detection, etc., capabilities which are desirable when testing a model for reading comprehension.

We chose datasets that adhere to two main properties: First, we exclude from consideration any multiple choice dataset, as these typically require very different model architectures, and they often have biases in how the distractor choices are generated. Second, we require that the dataset be originally designed for answering isolated questions over a single, given passage of text. We are focused on evaluating *reading* performance, not *retrieval*; reading a single passage of text is far from solved, so we do not complicate things by adding in retrieval, conversation state, or other similar complexities.

It is our intent to add to the evaluation server any high-quality reading comprehension dataset that is released in the future that matches these restrictions.

We now briefly describe the datasets that we include in the initial release of ORB. Table 7.1 gives summary statistics of these datasets. Except where noted, we include both the development and test sets (including hidden test sets) in our evaluation server for all datasets.

SQuAD [76] requires a model to perform lexical matching between the context and the question to predict the answer. This dataset provides avenues to learn predicate-argument structure and multi-sentence reasoning to some extent. It was collected by asking crowd-workers to create question-answer pairs from Wikipedia articles such that the answer is a single-span in the context. The dataset was later updated to include unanswerable questions [51], giving a harder question set without as many reasoning shortcuts. We include only the development sets of SQuAD 1.1 and SQuAD 2.0 in our evaluation server.

DuoRC [233] tests if the model can generalize to answering semantically similar but syntactically different paraphrased questions. The questions are created on movie summaries obtained from two sources, Wikipedia and IMDB. The crowd-workers formalized questions based on Wikipedia contexts and in turn answered them based on the IMDB context. This ensured that the model will not rely solely on lexical matching, but rather utilize semantic understanding. The answer can be either a single-span from context or free form text written by the annotator.

Quoref [116] focuses on understanding coreference resolution, a challenging aspect of natural language understanding. It helps gauge how a model can handle ambiguous entity and event resolution to answer a question correctly. This dataset was created by asking crowd workers to write questions and multi-span answers from Wikipedia articles that centered around pronouns in the context.

DROP [17] attempts to force models to have a more comprehensive understanding of a paragraph, by constructing questions that query many parts of the paragraph at the same time. These questions involve reasoning operations that are mainly rudimentary mathematical skills such as addition, subtraction, maximum, minimum, etc. To perform well on this dataset a model needs to locate multiple spans associated with questions in the context and perform a set of operations in a hierarchical or sequential manner to obtain the answer. The answer can be either a set of spans from the context, a number or a date.

ROPES [228] centers around understanding the implications of a passage of text, particularly dealing with the language of causes and effects. A system is given a background passage, perhaps describing the effects of deforestation on local climate and ecosystems, and a grounded situation involving the knowledge in the background passage, such as, *City A has more trees than City B*. The questions then require grounding the effects described in the background, perhaps querying which city would more likely have greater ecological diversity. This dataset can be helpful in

understanding how to apply the knowledge contained in a passage of text to a new situation.

NewsQA [234] dataset focuses on paraphrased questions with predicate-argument structure understanding. To some extent it is similar to DuoRC, however the examples are collected from news articles and offers diverse linguistic structures. This crowd-sourced dataset was created by asking annotators to write questions from CNN/DailyMail articles as context.

NarrativeQA [78] focuses on understanding temporal reasoning among various events that happen in a given movie plot. It also tests the model’s ability to “hop” between various parts of the context and not rely solely on sequential reasoning. The dataset was constructed by aligning books from Gutenberg ¹ with the summaries of their movie adaptations from various web resources. The crowd workers were asked to create complex questions about characters, narratives, events etc. from summaries and typically can be answered from summaries. In addition, crowd workers were required to provide answers that do not have high overlap with the context. In accordance with our format, we only use the version with the summaries as context in our evaluation server.

Dataset	Dev Size	Test Size	Avg. Context Len.	Avg. Answer Len.
SQuAD1.1	10,570	-	123.7	4.0
SQuAD2.0	10,570	-	127.5	4.2
DuoRC	12,233	13,449	1113.6	2.8
Quoref	2,418	2,537	348.2	2.7
DROP	9,536	9,622	195.1	1.5
ROPES	1,204	1,015	177.1	1.2
NewsQA	5,166	5,126	711.3	5.1
NarrativeQA	3,443	10,557	567.9	4.7

Table 7.1: Dataset Statistics

¹<http://www.gutenberg.org/>

7.3 Synthetic Augmentations

Prior works [89] have shown that RC models are brittle to minor perturbations in original dataset. Hence, to test the model’s ability to generalize to out-of-domain syntactic structures and be logically consistent in its answers, we automatically generate questions based on various heuristics. These heuristics fall in two broad categories.

1. The question is paraphrased to a minimal extent to create new syntactic structures, keeping the semantics of the question largely intact and without making any changes to the original context and answer.
2. The predicate-argument structures of a given question-answer pair are leveraged to create new WH-questions based on the object in the question instead of the subject. This rule-based method, adopted from [231], changes the question and answer keeping the context fixed.

We use five augmentation techniques, where the first four techniques fall into the first category and the last technique falls into the second category.

Invert Choice transforms a binary choice question by changing the order in which the choices are presented, keeping the answer the same.

More Wrong Choice transforms a binary choice question by substituting the wrong choice in the question with another wrong choice from the passage.

No Answer substitutes a name in the question for a different name from the passage to create with high probability a new question with no answer.

Dataset	IC	MWC	Imp	No-Ans	SEARs
NewsQA	0	0	501	347	16009
QuoRef	0	0	79	385	11759
DROP	1377	457	113	284	16382
SQuAD	16	0	875	594	28188
ROPES	637	119	0	201	2909
DuoRC	22	0	2706	-	45020

Table 7.2: Yields of augmented datasets

SEARs creates minimal changes in word selection or grammar while maintaining the original meaning of the question according to the rules described by [230].

Implication creates a new question-answer pair, where the object of the original question is replaced with the answer directly resulting in creation of a new WH-question where the answer is now the object of the original question. These transformations are performed based on rules described by [231].

We attempted all the above augmentation techniques on all the datasets (except NarrativeQA where entity and event tracking is complex and these simple transformations can lead to a high number of false positives). Table 7.2 shows the number of augmentations generated by each augmentation technique-dataset pair. A few sample augmentations are shown in Table 7.3.

After generating all the augmented datasets, we manually identified the augmentation technique-dataset pairs which led to high-quality augmentations. We sample 50 questions from each augmented dataset and record whether they satisfy the three criteria: (1) Is the question understandable, with little to no grammatical errors? (2) Is the question semantically correct? (3) Is the new answer the correct answer for the new question?

Table 7.4 shows the number of high-quality questions generated for each dataset. We keep the augmentation technique-dataset pairs where at least 90% of the question-answer pairs satisfy the above three criteria. We further test the performance of these augmentations (Section 4) on a

Template Type	Context (truncated)	Original QA Pair	Generated QA Pair
Invert Choice	... before halftime thanks to a David Akers 32-yard field goal, giving Detroit a 17-14 edge ... in the third, Washington was able to equalize with John Potter making his first career field goal from 43 yards out ... in the fourth, Detroit took the lead again, this time by way of Akers hitting a 28-yard field goal, giving Detroit a 20-17 lead...	Q: Which player scored more field goals, David Akers or John Potter? A: David Akers	Q: Which player scored more field goals, John Potter or David Akers? A: David Akers
More Wrong Choice	The first issue in 1942 consisted of denominations of 1, 5, 10 and 50 centavos and 1, 5, and 10 Pesos. ... 1944 ushered in a 100 Peso note and soon after an inflationary 500 Pesos note. In 1945, the Japanese issued a 1,000 Pesos note...	Q: Which year ushered in the largest Pesos note, 1944 or 1945? A: 1945	Q: Which year ushered in the largest Pesos note, 1942 or 1945? A: 1945
Implication	... In 1562, naval officer Jean Ribault led an expedition that explored Florida and the present-day Southeastern U.S., and founded the outpost of Charlesfort on Parris Island, South Carolina...	Q: When did Ribault first establish a settlement in South Carolina? A: 1562	Q: Who established a settlement in South Carolina in 1562? A: Ribault
No Answer	From 1975, Flavin installed permanent works in Europe and the United States, including ... the Union Bank of Switzerland, Bern (1996). ... The 1930s church was designed by Giovanni Muzio...	Q: Which permanent works did Flavin install in 1996? A: Union Bank of Switzerland, Bern	Q: Which permanent works did Giovanni Muzio install in 1996? A: No Answer
SEARs	... Dhul-Nun al-Misri and Ibn Wahshiyya were the first historians to study hieroglyphs, by comparing them to the contemporary Coptic language used by Coptic priests in their time...	Q: What did historians compare to the Coptic language? A: hieroglyphs	Q: What'd historians compare to the Coptic language? A: hieroglyphs

Table 7.3: Examples of generated augmentations with various templates.

BERT [32] based model to establish if the dataset has a sufficiently different question distribution from the original and has enough independent value to be incorporated into the evaluation server.

Dataset	IC	MWC	Imp	No-Ans	SEARs
NewsQA	-	-	47	47	50
QuoRef	-	-	45	48	50
DROP	46	42	36	48	50
SQuAD	15/16	-	47	48	50
ROPES	48	36	-	11	50
DuoRC	18/22	-	47	-	50

Table 7.4: Quality of augmented datasets (# of good questions out of 50 sampled)

7.4 Experiments

7.4.1 Model

We train a numerically-aware BERT-based model² (NABERT) on all the seven datasets and test its performance on existing datasets and synthetic augmentations. NABERT is a BERT based model with the ability to perform discrete operations like counting, addition, subtraction etc. We added support for “impossible” answers in the existing NABERT architecture by extending the answer type predictor which classifies the type of reasoning involved given a question into one of the following five categories: *number*, *span*, *date*, *count*, *impossible*. All the hyper-parameter settings were kept the same.

We noticed *catastrophic forgetting* on randomly sampling a minibatch for training, from all the datasets pooled together. To alleviate this problem, we sampled uniformly from each dataset in the beginning and then switched to sampling in proportion to the size of each dataset towards the end of the epoch [235]. This helped improve the performance on several dataset by 3-4% in EM, however, there is still a lot of room for improvement on this front. We also tried a simple BERT model [32] and MultiQA [229] but NABERT gave the best results on the seven development sets.

In case of DuoRC and NarrativeQA, some answers are free-form human generated and do not have an exact overlap with the context. However, the NABERT model is trained to predict a span’s start

²<https://github.com/raylin1000/drop-bert>

Dataset	Dev		Test		Narrative QA	BLEU-1	BLEU-4	METEOR	ROUGE
	EM	F ₁	EM	F ₁					
NewsQA	29.34	45.40	29.69	46.19	Dev Set	0.17	0.021	0.33	0.52
Quoref	34.49	42.65	30.13	38.39	Test Set	0.16	0.019	0.33	0.53
DROP	19.09	23.16	17.69	21.87					
SQuAD 1.1	68.03	78.55	-	-					
SQuAD 2.0	33.70	39.17	-	-					
ROPES	40.03	49.07	47.96	56.06					
DuoRC	25.65	34.28	23.44	31.73					

Table 7.5: Performance on baseline BERT model on different datasets

and end indices in the context. So for answers, which are not exact spans from the context we pick a span which has the highest ROUGE-L with the gold answer to serve as labels for training. However, for evaluation we use the original gold answer and not the extracted passage span for evaluating the model’s performance.

7.4.2 Existing Dataset Performance

Table 7.5 shows the result of evaluating on all of the development and test sets using our evaluation server. We chose the official metrics adopted by the individual datasets to evaluate the performance of our baseline model. As can be seen in the table, the results are quite poor, significantly below single-dataset state-of-the-art on all datasets. The training of our initial baseline appears to be dominated by SQuAD 1.1, or perhaps SQuAD 1.1 mainly tests reasoning that is common to all of the other datasets. Significant research is required to build reading systems and develop training regimes that are general enough to handle multiple reading comprehension datasets at the same time, even when all of the datasets are seen at training time.

7.4.3 Synthetic Augmentations

Table 7.6 shows the performance of the baseline model on various development sets and heuristically generated questions. The **More Wrong Choice** augmentation is omitted since a high enough quality and/or yield of questions could not be ensured for any of the datasets. When evaluated on out-of-domain linguistic structures, performance drops significantly for some augmentation-dataset pairs but only marginally for others. For questions generated by the **Invert Choice** augmentation, the model struggles to grasp the correct reasoning behind two answer options like *Art Euphoric or Trescott Street* and changes the prediction when the choices are flipped. However, relative to the dev set performances on the original datasets, the performance drop is almost nonexistent. For the **SEARs** based augmentation the generated linguistic variations are close to in-domain syntactic structure so we do not see much performance drop in most of the datasets except for ROPES and NewsQA. The **Implication** style questions create a large performance drop for NewsQA and SQuAD while having a performance boost for DuoRC. Finally, the **No-Ans** type questions have the worst performance across board for all datasets.

7.5 Related Work

Generalization and multi-dataset evaluation Recently there has been some work aimed at exploring the relation and differences between multiple reading comprehension datasets.

MULTIQA [229] investigates over ten RC datasets, training on one or more source RC datasets, and evaluating generalization, as well as transfer to a target RC dataset. This work analyzes the factors that contribute to generalization, and shows that training on a source RC dataset and transferring to a target dataset substantially improves performance. MultiQA also provides a single format including a model and infrastructure for training and comparing question answering datasets. We provide no training mechanism, instead focusing on very simple evaluation that is compatible with any training

	Dev		IC		Imp		No-Ans		SEARs	
	EM	F ₁	EM	F ₁	EM	F ₁	EM	F ₁	EM	F ₁
NewsQA	29.34	45.40	-	-	23.35	34.36	0.02	0.02	21.34	33.33
QuoRef	34.49	42.65	-	-	32.91	44.84	0.0	0.0	34.84	42.11
DROP	19.09	23.16	40.23	48.03	-	-	0.0	0.0	16.97	21.65
SQuAD	68.03	78.55	56.25	64.58	46.74	57.97	0.0	0.0	56.53	71.25
ROPES	40.03	49.07	24.08	31.74	-	-	-	-	14.05	19.12
DuoRC	25.65	34.28	27.27	34.19	30.30	35.23	-	-	21.51	28.85

Template Type	Answered Incorrectly	Answered Correctly
Invert Choice	<p>Original: Which art gallery was founded first, Art Euphoric or Trescott Street?</p> <p>Generated: Which art gallery was founded first, Trescott Street or Art Euphoric?</p>	<p>Original: Who scored more field goals, Nate Kaeding or Dan Carpenter?</p> <p>Generated: Who scored more field goals, Dan Carpenter or Nate Kaeding?</p>
Implication	<p>Original: When did the Huguenots secure the right to own land in the Baronies?</p> <p>Generated: Who secured the right to own land in baronies in 1697?</p>	<p>Original: When did Henry issue the Edict of Nantes?</p> <p>Generated: What did Henry issue in 1598?</p>
SEARs	<p>Original: What was the theme of Super Bowl 50?</p> <p>Generated: So what was the theme of Super Bowl 50?</p>	<p>Original: Who won Super Bowl 50?</p> <p>Generated: So who won Super Bowl 50?</p>

Table 7.6: Quantitative and qualitative analysis of generated augmentations. We only show performance for high yield and high-quality augmentations.

regime, including evaluating on hidden test sets.

MRQA19, the Machine Reading for Question Answering workshop, introduced a shared task, which tests whether existing machine reading comprehension systems can generalize beyond the datasets on which they were trained. The task provides six large-scale datasets for training, and evaluates generalization to ten different hidden test datasets. However these datasets were modified from their original version, and context was limited to 800 tokens. In addition this shared task only tests for generalization with no intra-domain evaluation. In contrast, our evaluation server simply provides a single-model evaluation on many different datasets, with no prescriptions about training

regimes.

NLP evaluation benchmarks The General Language Understanding Evaluation benchmark or GLUE [236] is a tool for evaluating and analyzing the performance of models across a diverse range of existing NLU tasks. A newer version, Super-GLUE [82] is styled after GLUE with a new set of more difficult language understanding tasks. In this line of work another standard toolkit for evaluating the quality of universal sentence representations is SENTEval [237]. Similar to GLUE, SENTEval also encompasses a variety of tasks, including binary and multi-class classification, natural language inference and sentence similarity. We differ from GLUE and SENTEval by focusing on reading comprehension tasks, and only evaluating a single model on all datasets, instead of allowing the model to be tuned to each dataset separately.

Evaluation Platforms and Competitions in NLP The use of online evaluation platform with private test labels has been exercised by various leaderboards on Kaggle and CodaLab, as well as shared tasks at the SemEval and CoNLL conferences.

Additional benchmarks such as PARLAI [238] and BABI [239] proposed a hierarchy of tasks towards building question answering and reasoning models and language understanding. However these frameworks do not include a standardized evaluation suite for system performance nor do they offer a wide set of reading comprehension tasks.

7.6 Conclusion

We have presented ORB, an open reading benchmark designed to be a comprehensive test of reading comprehension systems, in terms of their generalizability, understanding of various natural language phenomenon, capability to make consistent predictions, and ability to handle out-of-

domain questions. This benchmark will grow over time as more interesting and useful reading comprehension datasets are released. We hope that this benchmark will help drive research on general reading systems.

Chapter 8

Dynamic Sampling for improving multi-task generalization

A single observation that is inconsistent with some generalization points to the falsehood of generalization, and thereby points to itself.

Ian Hacking

(Canadian Philosopher)

Building general reading comprehension systems, capable of solving multiple datasets at the same time, is a recent aspirational goal in the research community. Prior work has focused on model architectures or generalizations to held-out datasets and largely passed over the particulars of the multi-task learning setup, which focuses on learning a general-purpose single model. We show that a simple dynamic sampling strategy, selecting instances for training proportional to the multi-task model's current performance on a dataset relative to its single-task performance, gives substantive gains over prior multi-task sampling strategies, mitigating the catastrophic forgetting

that is common in multi-task learning. We also demonstrate that allowing instances of different tasks to be interleaved as much as possible between each epoch and batch has a clear benefit in multi-task performance over forcing task homogeneity at the epoch or batch level. Our final model shows greatly increased performance over the best model on ORB, a multi-task reading comprehension benchmark.

8.1 Introduction

Building multi-task reading comprehension systems has received significant attention and been a focus of active research [240, 241]. These approaches mostly focus on model architecture improvements or generalizability to new tasks or domains. While these contributions are important, it is also important to explore the optimal way to structure training; as we will show, training on instances from diverse datasets (tasks) means that unlike in a single-task setting, ample instances from each task distribution must be represented during training to properly capture that diversity. We explore 2 fundamental aspects of structuring multi-task training: how many instances are sampled from each task per epoch and how those instances are organized within the epoch. We investigate the importance of this structuring by training a multi-task model on the 8 datasets from ORB [15], a recent multi-task reading comprehension benchmark.

We first explore the sampling distribution over datasets at each epoch: how many instances from each dataset should be used to train. Prior work has typically either made this a uniform distribution over datasets (implicitly favoring smaller datasets), a distribution proportional to the sizes of the datasets (implicitly favoring larger datasets), or some combination of the two. Because these sampling strategies favor some datasets over others, they can lead to catastrophic forgetting in the non-favored datasets. We introduce a *dynamic sampling strategy* that selects instances from a dataset with probability proportional to the gap between its current performance on some metric (like EM or F1 score) and measured single-task performance of the same model on that dataset.

By adjusting the sampling distribution over the course of training according to what the model is learning, this method is able to mitigate the catastrophic forgetting that is observed with other sampling strategies.

Next we explore the impact of within-epoch scheduling strategies: once a set of instances has been selected for training, how should they be ordered and batched together? We explore three different strategies: partitioning, homogeneous batches, and heterogeneous batches. We observe a steady increase in performance as instances from different datasets become more and more interleaved within an epoch (less partitioned) and batches are more heterogeneous. This suggests that more variety in batches aids convergence when performing gradient descent steps as opposed to steps using homogeneous batches which only update the model with respect to one task at a time. Partitioning also yields poorer performance since it does not allow the model to see the least recent tasks later in the epoch which leads to catastrophic forgetting on those tasks.

We empirically evaluate these various training strategies on ORB, a recent multi-task reading comprehension benchmark: we take the previous best published model and retrain it using dynamic sampling and heterogeneous batches, leading to a performance increase averaging about 12 points EM and 9 points F1 per task. While we only evaluate on reading comprehension, the methods we present are quite general and can be applied to any multi-task learning setting.

8.2 Sampling and Scheduling Strategies

We explore two main dimensions along which the instances are ordered in multi-task learning: (1) *instance sampling* from each dataset to get a collection of examples to use for an epoch; and (2) *within-epoch scheduling* of those instances, determining how they should be ordered and batched. The key consideration for these various strategies is avoiding a phenomenon similar to “catastrophic forgetting” [242], where performance on a specific dataset in an unbalanced training set can drop

dramatically when training moves on from that dataset.

8.2.1 Instance Sampling

We investigate the following four alternatives for determining how many instances to draw from each dataset for each epoch:

Uniform The simplest way is to uniformly sample instances for each task [99], which results in an approximately equal number of instances from each dataset per epoch. In practice, this means randomly sampling the same number of training instances from each dataset at each epoch, which will likely be a small subset of all the training instances, as the number of instances is constrained by the smallest dataset. Large datasets will be proportionally underrepresented here.

By Size Alternatively, unbalanced datasets can be dealt with by sampling from each task in proportion to their training set size [e.g. 243]. However, this approach can result in underfitting small-sized tasks and overfitting large-sized tasks if the ratio between size differences is too extreme.

Uniform→Size¹ This sampling scheme simply has instances sampled uniformly for the first half of training epochs and has instances sampled by training set size for the second half.

Dynamic The prior two methods use a fixed sampling distribution for every epoch of training. We introduce a new, dynamic sampling strategy that aims to focus training on places where it is most needed. For this sampling strategy, we first compute single-task validation metrics for the model that we are training. For each task, we calculate the gap between current multi-task performance and the respective single-task performance and normalize these metric differentials to create a

¹github.com/mrqa/MRQA-Shared-Task-2019

probability distribution. Then, for every epoch after the first (where we use sampling by size), we sample instances by task from this distribution.

If performance on a dataset is far from single-task performance, it will get sampled heavily, while datasets that have reached or exceeded single-task performance will get sampled little if at all.²

We also experimented with modifying the metric used to calculate the differential. We tested using the (1) validation loss differential (2) validation EM differential (3) validation F1 differential and (4) the sum of the validation EM and F1 differentials.

Amongst these, the validation loss for each dataset reaches the single-task loss far quicker than others. This is likely due to the phenomenon that neural networks can overfit to specific loss functions while still benefitting in terms of accuracy [245]. This explains why the gap in accuracy metrics can be so wide while the loss gap closed within 1 or 2 epochs. Because of this behavior, the loss differentials were all nearly identical in the first few epochs and behavior became very similar to uniform sampling. We finally decided to use EM+F1 differential as this yielded nominally better performance than EM or F1 differential and significantly better performance than loss differential.

8.2.2 Epoch Scheduling

We explore several different methods for scheduling and batching the instances within an epoch after the set of instances has been sampled:

Partitioned This scheduling strategy partitions the instances in the epoch by task. In other words, the model will never see an instance from a new dataset until all the instances from the current dataset are exhausted. It seems intuitive that this strategy would exacerbate catastrophic forgetting on the tasks it saw least recently, especially when there are a large number of tasks. We include this

²[244] use a related technique in reinforcement learning, though the setup is different.

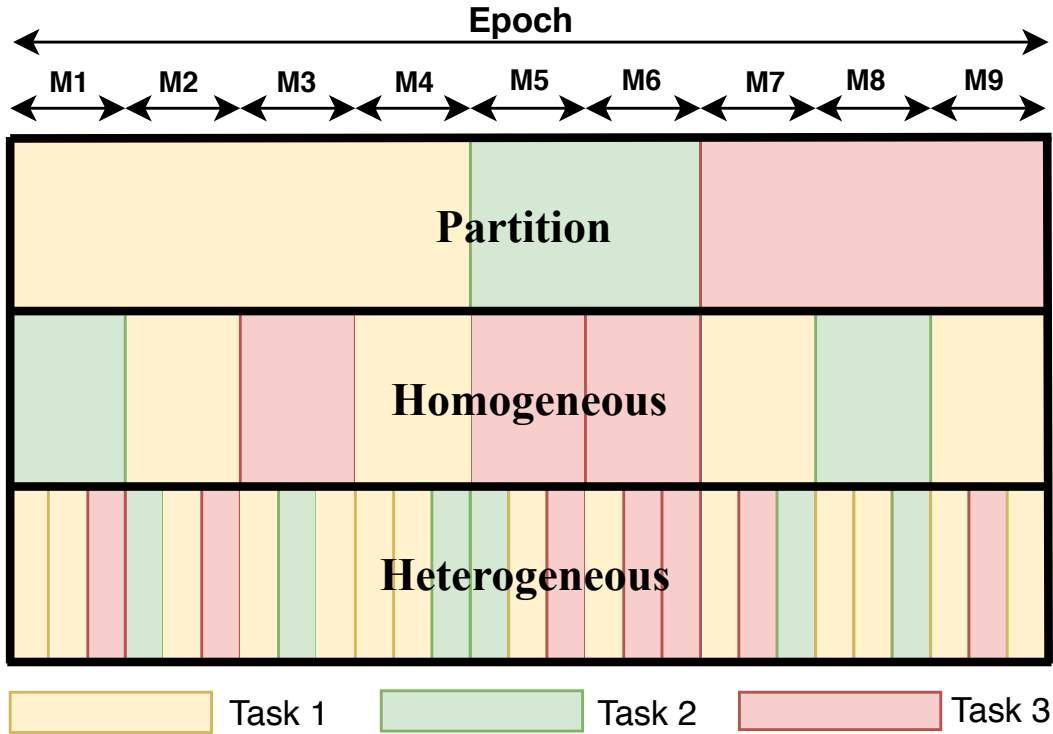


Figure 8.1: **Illustration of Epoch Scheduling Strategies with Dynamic Sampling.** Instances are sampled dynamically in proportion to exact match accuracy difference of 25%, 10% and 15% for task 1, 2 and 3 respectively. M1, M2, ... M9 depict nine mini-batches in an epoch.

method simply for completeness.

Homogeneous Batches This scheduling strategy does not force instances into partitions based on the dataset. Instead, instances from each dataset are batched together, then the batches are shuffled.

Heterogeneous Batches This scheduling strategy shuffles all selected instances for the epoch, then batches them together. Each batch could have instances from many different datasets.

Uniform Batches This scheduling strategy is used by the baseline model for the MRQA shared task [200] as well as for the best prior result on ORB. This method places one instance per dataset in each batch (forced heterogeneity) until the smallest dataset runs out of instances. This strategy continues with the remaining datasets, until all datasets are exhausted.

Dataset	Train Size	Dev Size
Small		
Quoref	19,392	2,407
ROPES	10,924	1,688
Medium		
DuoRC	54,746	12,224
NarrativeQA	32,717	3,393
Large		
DROP	77,394	9,530
NewsQA	92,543	5,154
SQuAD1.1	87,596	10,570
SQuAD2.0	130,310	11,864

Table 8.1: Open Reading Benchmark (ORB) Datasets

Method	Average	Quoref	ROPES	DuoRC	NarrQA	SQuAD	SQuAD2	DROP	NewsQA
	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁
Single Task	- -	53.0 58.6	67.5 72.1	23.3 30.8	- 50.3	57.5 73.5	66.0 69.6	57.1 54.4	35.3 49.8
Uniform	49.2 55.8	56.9 61.5	69.7 74.3	23.4 32.1	- 53.1	69.3 78.0	38.1 42.9	51.8 54.4	35.0 49.9
By Size	50.0 56.3	53.7 57.7	62.7 68.1	23.3 31.6	- 52.4	65.8 74.1	58.1 63.0	52.0 54.5	34.6 49.1
Uni→Size	49.7 56.5	55.8 60.0	68.8 73.8	23.2 32.0	- 53.0	52.0 63.7	63.4 67.4	49.7 52.2	35.0 49.8
Dynamic	51.7 58.1	56.3 60.4	65.1 71.9	23.1 31.5	- 52.9	66.3 74.7	63.2 67.7	53.8 56.3	34.5 49.2

Table 8.2: Effect of using different instance sampling strategies with heterogeneous batch scheduling

8.3 Experiments

Setup The eight reading comprehension tasks are from the ORB benchmark [15]: DROP [17], DuoRC [233], NarrativeQA [246], NewsQA [234], Quoref [116], ROPES [228], SQuAD [76], and SQuAD 2.0 [51].

We use the NABERT³ (Numerically Augmented BERT) model with an additional reasoning type to allow “No Answer” as an answer to accommodate the SQuAD 2.0 dataset which has about 40,000 “No Answer” questions. Each training session lasted 30 epochs with 50,000 instances

³https://github.com/raylin1000/drop_bert

Method	Average	Quoref	ROPES	DuoRC	NarrQA	SQuAD	SQuAD2	DROP	NewsQA
	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁
Partition	46.1 53.2	50.7 55.3	58.1 65.4	22.1 30.7	- 50.9	67.0 76.6	36.5 41.6	55.3 58.2	32.0 47.4
Homo	48.8 54.7	53.3 56.8	61.5 66.6	21.6 29.6	- 49.9	63.7 71.7	56.0 60.6	51.8 54.1	33.5 48.2
Hetero	51.7 58.1	56.3 60.4	65.1 71.9	23.1 31.5	- 52.9	66.3 74.7	63.2 67.7	53.8 56.3	34.5 49.2

Table 8.3: Effect of using different epoch scheduling strategies with dynamic sampling

Method	Average	Quoref	ROPES	DuoRC	NarrQA	SQuAD	SQuAD2	DROP	NewsQA
	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁	EM F ₁
ORB	34.4 42.1	35.0 44.7	31.1 37.3	25.4 34.1	- 36.6	67.3 77.7	32.8 38.0	20.2 23.6	29.2 44.6
Dynamic	47.6 54.5	59.4 63.9	36.5 44.8	23.0 31.5	- 52.0	66.3 74.7	61.2 65.7	51.9 54.2	34.7 49.1

Table 8.4: Results on ORB test sets.

sampled per epoch. Three training sessions were conducted per sampling method and the EM and F1 scores shown are averaged over those three sessions. Note that NarrativeQA is evaluated using only ROUGE F1 score. Due to GPU memory constraints, we are limited to a batch size of 4, so we are unable to replicate the *Uniform Batches* configuration of MRQA (requires a batch size of 8 to fit 1 instance from each of the 8 datasets).

Sampling Strategies Table 8.2 shows the effectiveness of the sampling techniques discussed above. Uniform sampling yields a very mediocre performance for 7 datasets but significantly underperforms on SQuAD 2.0, which is likely not getting enough representation each epoch for its unique no-answer questions. Sampling by size yields mediocre performances for 7 datasets but underperforms on ROPES, which is easily the smallest dataset and therefore gets undersampled. However, performance on Quoref, the second smallest dataset, is still relatively high, which might be explained by its SQuAD-style questions. Exposure to SQuAD, one of the largest datasets, likely benefits performance on Quoref as well. Interestingly, uniform sampling followed by size sampling slightly alleviates the problems from the individual sampling methods but also slightly underperforms on DROP. Finally, dynamic sampling achieves the highest average performance and fully cures both

problems mentioned above since each epoch, the sampling distribution can be adjusted based on which datasets perform poorly. The previous sampling methods have static sampling distributions, so these adjustments are impossible.

Scheduling Strategies Table 8.3 show that heterogeneous batches during sampling leads to the best multi-task performance, and performance steadily decreases as instance grouping becomes more and more homogenized with respect to the dataset.

ORB Evaluation Finally, Table 8.4 shows that our model trained with dynamic sampling and heterogeneous batches significantly outperforms the previous ORB state-of-the-art NABERT baseline model (submitted on 11/12/2019 on the leaderboard site⁴).

8.4 Conclusions

Our goal was to investigate which instance sampling method and epoch scheduling strategy gives optimal performance in a multi-task reading comprehension setting. The results suggest that dynamic sampling—sampling instances from each task based on their respective metric differentials—is a fruitful direction to explore for improving performance. We also show that interleaving instances from different tasks within each epoch and forming heterogeneous batches is crucial for optimizing multi-task performance. It is also worth noting that for the DuoRC, NarrativeQA, SQuAD, and Quoref datasets there are cases where the multi-task model outperforms the single-task model. This suggests that for specific cases, we observe an effect similar to data augmentation (like exposure to SQuAD benefitting QuOREF performance as mentioned above) but this needs to be explored further. We hope that future work experiments further with dynamic sampling such as by modifying the metric (e.g., using BLEU or ROUGE score if applicable) and/or modifying other values like number

⁴<https://leaderboard.allenai.org/orb/submissions/public>

of instances per epoch based on performance metrics (not only does this effectively change learning rate, but it would also allow the model to update the sampling distribution more or less frequently).

Chapter 9

Benefits of Intermediate Annotations

A goal without a plan is just a wish.

Antoine de Saint-Exupéry

(French writer and pioneering aviator)

Complex, compositional reading comprehension datasets require performing latent sequential decisions that are learned via supervision from the final answer. A large combinatorial space of possible decision paths that result in the same answer, compounded by the lack of intermediate supervision to help choose the right path, makes the learning particularly hard for this task. In this work, we study the benefits of collecting intermediate reasoning supervision along with the answer during data collection. We find that these intermediate annotations can provide two-fold benefits. First, we observe that for any collection budget, spending a fraction of it on intermediate annotations results in improved model performance, for two complex compositional datasets: DROP and Quoref. Second, these annotations encourage the model to learn the correct latent reasoning steps, helping combat some of the biases introduced during the data collection process.

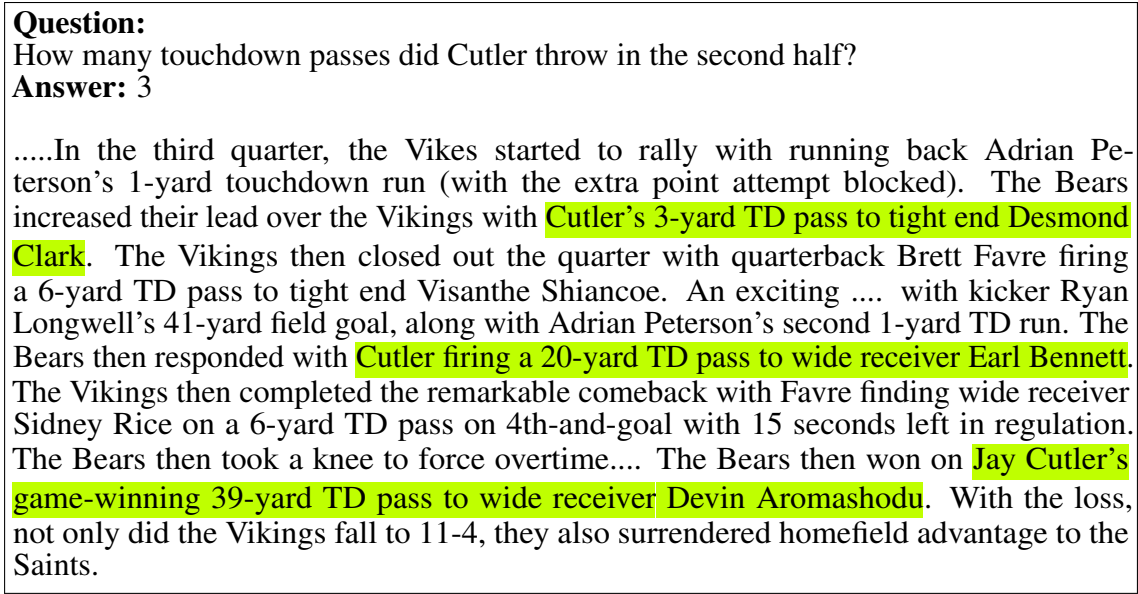


Figure 9.1: Example from DROP, showing the intermediate annotations that we collected via crowd-sourcing.

9.1 Introduction

Recently many reading comprehension datasets requiring complex and compositional reasoning over text have been introduced, including HotpotQA [40], DROP [247], Quoref [116], and ROPES [228]. However, models trained on these datasets [193, 180] only have the final answer as supervision, leaving the model guessing at the correct latent reasoning. Figure 9.1 shows an example from DROP, which requires first locating various operands (i.e. relevant spans) in the text and then performing filter and count operations over them to get the final answer “3”. However, the correct answer can also be obtained by extracting the span “3” from the passage, or by adding or subtracting various numbers in the passage. The lack of intermediate hints makes learning challenging and can lead the model to rely on data biases, limiting its ability to perform complex reasoning.

In this paper, we present three main contributions. First, we show that annotating relevant context spans, given a question, can provide an easy and low-cost way to learn better latent reasoning. To be precise, we show that under low budget constraints, collecting these annotations for up to 10% of the training data (2-5% of the total budget) can improve the performance by 4-5% in F1.

We supervise the current state-of-the-art models for DROP and Quoref, by jointly predict the relevant spans and the final answer. Even though these models were not designed with these annotations in mind, we show that they can still be successfully used to improve model performance. Models that explicitly incorporate these annotations might see greater benefits. Our results suggest that future dataset collection efforts should set aside a fraction of budget for intermediate annotations, particularly as the reasoning required becomes more complex.

Second, these annotations can help combat biases that are often introduced while collecting data [139, 138]. This can take the form of label bias—in DROP, 18% of questions have answers 1, 2, or 3—or annotator bias, where a small group of crowd workers creates a large dataset with common patterns. By providing intermediate reasoning steps explicitly, the annotations we collect help the model overcome some of these biases in the training data.

Finally, the intermediate annotations collected in this work, including 8,500 annotations for DROP and 2,000 annotations for Quoref, will be useful for training further models on these tasks. We have made them available at https://github.com/dDua/Intermediate_Annotations.

9.2 Intermediate Annotations

Intermediate annotations describe the right set of context spans that should be aggregated to answer a question. We demonstrate their impact on two datasets: DROP and Quoref. DROP often requires aggregating information from various events in the context (Figure9.1). It can be challenging to identify the right set of events directly from an answer when the same answer can be derived from many possible event combinations. We annotate the entire event span including all the attributes associated with the specific event. Quoref requires understanding long chains of coreferential reasoning, as shown in Figure9.2, which are often hard to disentangle, especially when the context refers to multiple entities. We specifically annotate the coreference chains which lead to the entity

Question:
 What record do the children that Conroy teaches play back to him?
Answer: Beethoven’s Fifth Symphony

Conroy tries to teach them about the outside world but comes into conflict both with the principal and Mr. Skeffington, the superintendent. **He** teaches them how to brush their teeth, who Babe Ruth is, and has the children listen to music, including Flight of the Bumblebee and Beethoven’s Fifth Symphony. **He** explains that when Beethoven wrote the Fifth Symphony, **he** was writing about “what death would sound like”. **He** is also astounded they’ve never even heard of Halloween, and **he** decides to take them to Beaufort on the mainland to go trick-or-treating, which the superintendent has forbidden. **He** also must overcome parental fears of “the river.” As **he** leaves the island for the last time, the children come out to see **him** leave, all of them lined up on a rickety bridge. As **he** is about to leave by boat, **one of the students then begins playing a record, which is the beginning movement of Beethoven’s Fifth Symphony.**

Figure 9.2: Example collected annotation from Quoref, showing the intermediate steps.

being queried.

Collection process: We used Amazon Mechanical Turk to crowd-source the data collection.

We randomly sample 8,500 and 2,000 QA pairs from the training set for DROP and Quoref respectively. We showed a QA pair and its context to the workers and asked them to highlight “essential spans” in the context. In case of DROP, crowd workers were asked to highlight complete events with all their corresponding arguments in each span. For Quoref, they were asked to highlight the coreference chains associated with the answer entity in the context.

Cost of gathering intermediate annotations: Each HIT, containing ten questions, paid \$1, and took approximately five minutes to complete. Overall, we spent \$850 to collect 8,500 annotations for DROP and \$200 to collect 2,000 annotations for Quoref. If these annotations are collected simultaneously with dataset creation, it may be feasible to collect them at a lower cost, as the time taken to read the context again will be avoided.

9.3 Experiments and Results

In this section, we train multiple models for the DROP and Quoref datasets, and evaluate the benefits of intermediate annotations as compared to traditional QA pairs. In particular, we will focus on the cost vs benefit tradeoff of intermediate annotations, along with evaluating their ability to mitigate bias in the training data.

9.3.1 Setup

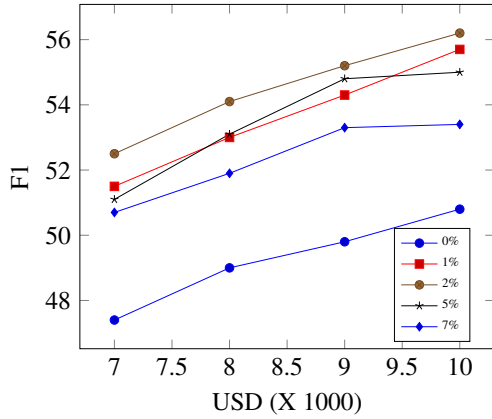
We study the impact of annotations on DROP on two models at the top of the leaderboard: NABERT¹ and MTMSN [193]. Both the models employ a similar arithmetic block introduced in the baseline model [247] on top of contextual representations from BERT [32]. For Quoref, we use the baseline XLNet [248] model released with the dataset.

We supervise these models with the annotations in a simple way, by jointly predicting intermediate annotation and the final answer. We add two auxiliary loss terms to the marginal log-likelihood loss function. The first is a cross-entropy loss between the gold annotations (g) and predicted annotations, which are obtained by passing the final BERT representations through a linear layer to get a score per token p , then normalizing each token’s score of being selected as an annotation with a sigmoid function.

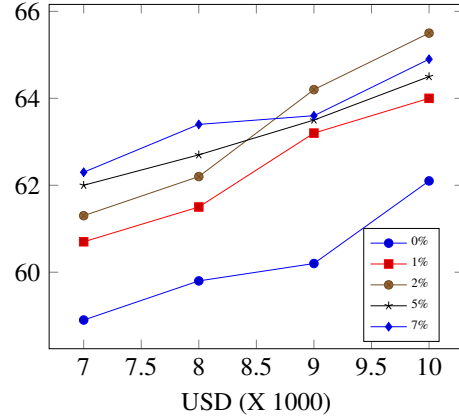
$$\mathcal{L}_1(\theta) = \alpha_1 CE(g, \sigma(p)) \tag{9.1}$$

The second is an L_1 loss on the sum of predicted annotations, encouraging the model to only select

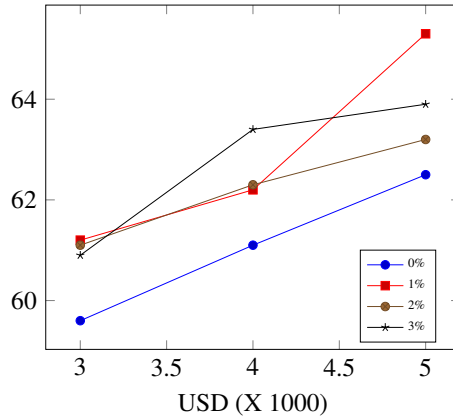
¹https://github.com/raylin1000/drop_bert



(a) Fixed cost: NABERT DROP



(b) Fixed cost: MTMSN DROP



(c) Fixed cost: Quoref XLNet

Figure 9.3: Performance of model for a varying percentage of budget invested in collecting intermediate annotation. The calculation was done with costs as \$0.4 and \$0.7 for a QA pair in DROP and Quoref, respectively.

a subset of the passage.

$$\mathcal{L}_2(\theta) = \alpha_2 \sum_{\ell=0}^{|\text{tokens}|} \sigma(p_\ell)$$

The hyper-parameters α_1 and α_2 were used to balance the scale of both auxiliary loss terms with the marginal log-likelihood.

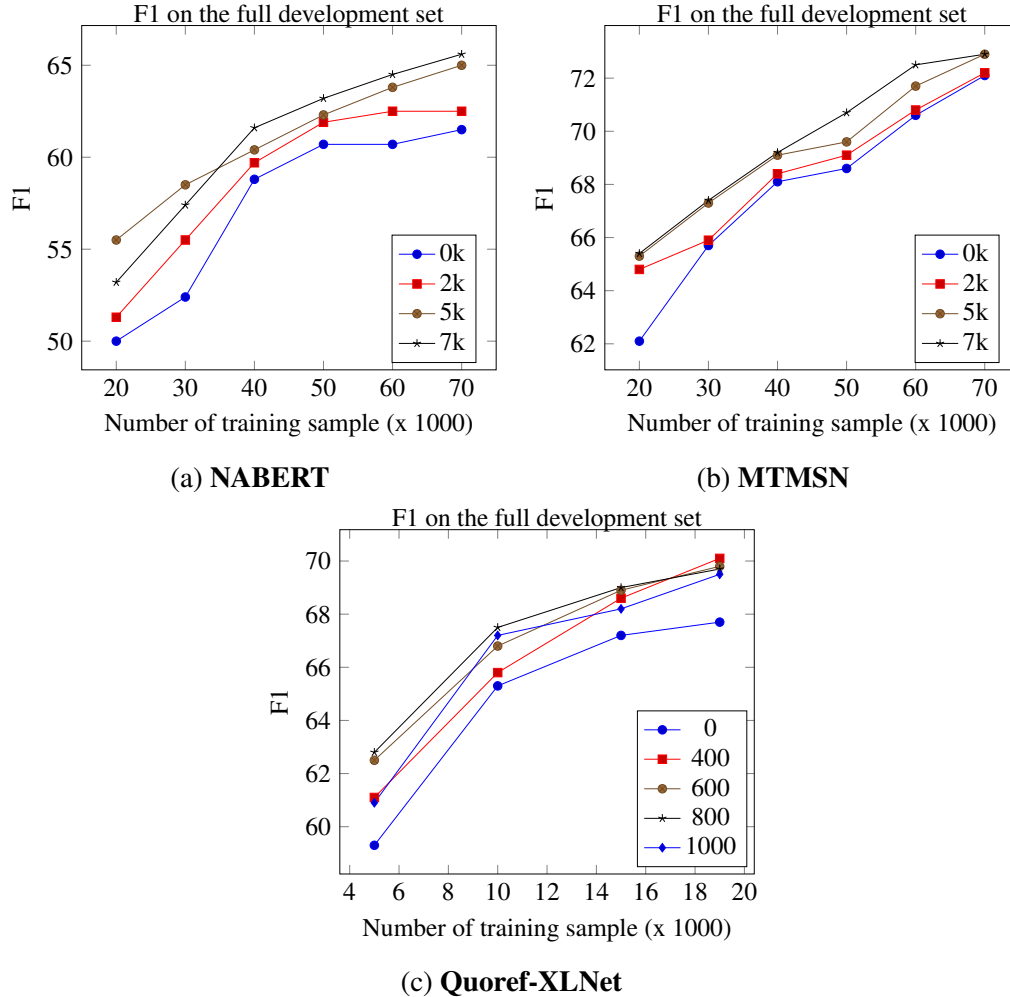


Figure 9.4: Performance of model trained on varying amount of annotations used in training

9.3.2 Cost vs Benefit

To evaluate the cost-benefit trade-off, we fix the total collection budget and then vary the percentage of budget that should go into collecting intermediate annotations. As shown in Figure9.3a, the model achieves better performance (+1.7% F1) when spending \$7k where 2% budget is used for collecting intermediate reasoning annotations as compared to model performance when spending \$10k for collecting only QA pairs. Overall, from Figure9.3 we can see that allocating even 1% of the budget to intermediate annotations provides performance gains. However, we observe that allocating a large percentage of the budget to intermediate annotations at the expense of QA pairs reduces performance. In our experiments, we find that the sweet-spot percentage of the budget and

training-set that should be allocated to intermediate annotations is 2% and ~10% respectively.

9.3.3 Bias Evaluation

Unanticipated biases [140, 249] are often introduced during dataset collection due to many reasons (eg., domain-specific contexts, crowd-workers distributions, etc.). These “dataset artifacts” can be picked up by the model to achieve better performance without learning the right way to reason. We explore two examples of such dataset artifacts in DROP and Quoref.

In DROP, around 40% of the passages are from NFL game summaries. The frequency of counting and arithmetic questions from this portion of the data resulted in the answers 1, 2, and 3 making up 18% of the entire training set.

To study the effect of biased answer distribution on model performance, we sample 10k QA pairs with answers $\in [0,9]$ from the training set *randomly* as a biased training set. We also sample QA pairs from the validation set *uniformly* for each answer $\in [0,9]$ thus ensuring that each answer has equal representation in the unbiased validation set.

In Quoref, we found that around 65% of the answers are entity names present in the first sentence of the context. Similar to DROP, we create a biased training set with 5k QA pairs from the original training data, and an unbiased validation set with equal representation of answers from the first sentence and the rest of the context.

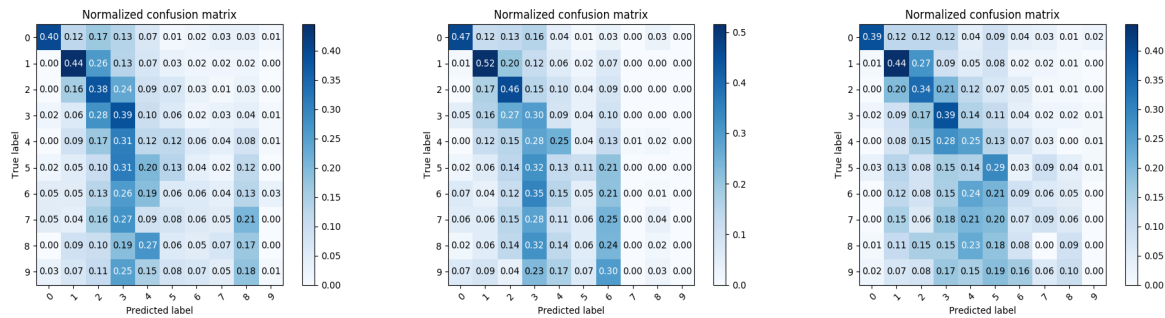
We investigate the effects of spending a small additional budget, either by adding more QA pairs (from the biased data distribution) or by collecting intermediate annotations, on this bias.

We use two metrics to measure the extent to which bias has been mitigated. The first is the original metric for the task, i.e. F_1 , that measures how accurate the model is on the unbiased evaluation. Further, we also want to evaluate the extent to which the errors made by the model are unbiased;

Dataset	Baseline		More QA pairs		Annotations	
	F1 (%)	Conf. loss	F1 (%)	Conf. loss	F1 (%)	Conf. loss
DROP	24.6	101.5	25.5	107.5	28.1	94.5
Quoref	61.8	103.0	62.7	109.0	64.3	97.0

Table 9.1: F1 performance and confusion loss (lower is better) of models in three settings: baseline with 10k(DROP) and 5k(Quoref) QA pairs, additional QA pairs worth \$250 and \$100 for DROP and Quoref respectively, and additional annotations worth \$250 and \$100 for DROP and Quoref respectively. To put confusion loss in perspective, the *best* confusion loss, i.e. perfect diffusion, is 90.1 for DROP and 87.0 for Quoref.

in other words, how much is the error *diffused* over all possible answers, rather than only over the biased labels. We compute *confusion loss* [250] as the metric for this, which measures error diffusion by computing the highest singular value of the unnormalized confusion matrix after setting the diagonal elements (i.e. true positives), to zero [251] (lower confusion loss implies more diffusion). In an ideal scenario, all labels should have an equally likely probability of being a mis-prediction. Higher confusion loss implies that if we consider mis-classifications of a model we see that it has a tendency of over-predicting a specific label, making it biased towards that specific class.



(a) 10k samples (b) Additional QA pairs worth \$250 (c) Annotations worth \$250

Figure 9.5: For the same cost intermediate annotations helps diffuse biased over-representation of number 3 as compared to adding more question-answer pairs

Table 9.1 shows that along with higher improvements in F_1 on providing annotations as compared to more QA pairs, we also see a reduction in the confusion loss with annotations indicating bias mitigation.

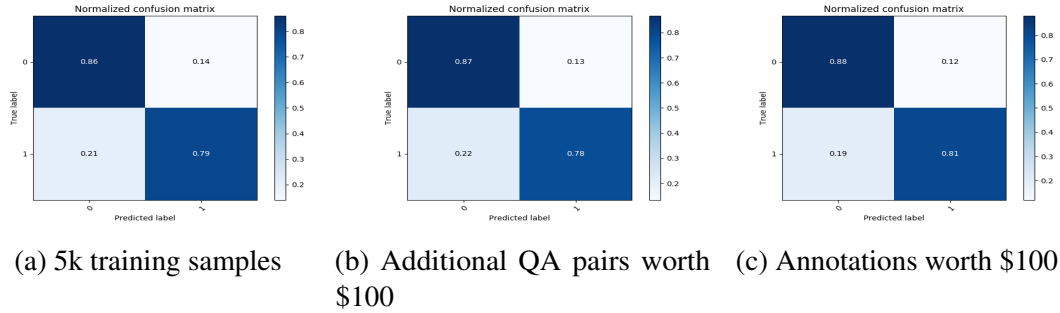


Figure 9.6: For the same cost intermediate annotations helps diffuse biased over-representation of number 3 as compared to adding more question-answer pairs

How many times did the Cowboys score in the first half?

Still searching for their first win, the Bengals flew to Texas Stadium for a Week 5 interconference duel with the Dallas Cowboys. In the first quarter, Cincinnati trailed early as Cowboys kicker Nick Folk got a 30-yard field goal, along with RB Felix Jones getting a 33-yard TD run. In the second quarter, Dallas increased its lead as QB Tony Romo completed a 4-yard TD pass to TE Jason Witten. The Bengals would end the half with kicker Shayne Graham getting a 41-yard and a 31-yard field goal. In the third quarter, Cincinnati tried to rally as QB Carson Palmer completed an 18-yard TD pass to WR T. J. Houshmandzadeh. In the fourth quarter, the Bengals got closer as Graham got a 40-yard field goal, yet the Cowboys answered with Romo completing a 57-yard TD pass to WR Terrell Owens. Cincinnati tried to come back as Palmer completed a 10-yard TD pass to Houshmandzadeh (with a failed 2-point conversion), but Dallas pulled away with Romo completing a 15-yard TD pass to WR Patrick Crayton.

Figure 9.7: Predicted relevant spans for question answered correctly with annotation (prediction: “3”) and incorrectly without annotations (prediction: “2”) by MTMSN model trained on DROP

Further, we also find that for DROP, the false positive rate for top-3 common labels fell from 47.7% (baseline) to 39.6% (with annotations), while the false positive rate for the bottom-7 increased from 30.4%(baseline) to 36.3%(with annotations), further demonstrating mitigation of bias. The confusion matrices are included in Appendix.

9.3.4 Qualitative Result

Figure 9.7 shows a DROP example where the model trained without annotations is not able to determine the right set of events being queried, returning an incorrect response. The model trained

What is the full name of Mary Harriette’s father?

Motteux was also without heirs and bequeathed Sandringham, together with another Norfolk estate and a property in Surrey, to the third son of his close friend, Emily Lamb, the wife of Lord Palmerston. At the time of his inheritance in 1843, **Charles Spencer Cowper** was a bachelor diplomat, resident in Paris. On succeeding to Motteux’s estates, he sold the other properties and based himself at Sandringham. He undertook extensions to the hall, employing Samuel Sanders Teulon to add an elaborate porch and conservatory. **Cowper’s** style of living was extravagant **he and his wife** spent much of **their** time on the Continent and within 10 years the estate was mortgaged for £89,000. **The death of their only child, Mary Harriette**, from cholera in 1854 led the couple to spend even more time abroad, mainly in Paris, and by the early 1860s Cowper was keen to sell the estate.

Figure 9.8: Predicted relevant spans for question answered correctly with annotation (prediction:“Charles Spencer Cowper”) and incorrectly without annotations (prediction:“Lord Palmerston”) by XLNet on Quoref

with annotations can understand the semantics behind the query terms “first half” and “Cowboys”, to arrive at the correct answer. The curves depicting quantitative performance gains with varying amounts of annotations and QA pairs are in the appendix.

9.3.5 Related Works

Similar to our work, [252] studied the impact of providing explicit supervision via *rationales*, rather than generating them, for varying fractions of training set in text classification. However, we study the benefits of such supervision for complex compositional reading comprehension datasets. In the field of computer vision, [253] collected similar annotations, for visual recognition, where crowd-workers highlighted relevant regions in images.

Within reading comprehension, various works like HotpotQA [40] and CoQA [254] have collected similar reasoning steps for entire dataset. Our work shows that collecting intermediate annotations for a fraction of dataset is cost-effective and helps alleviate dataset collection biases to a degree. Another line of work [255] explores the cost vs. benefit of collecting full vs. partial annotations for various structured predictions tasks. However, they do not focus on intermediate reasoning required to learn the task.

Our auxiliary training with intermediate annotations is inspired by extensive related work on training

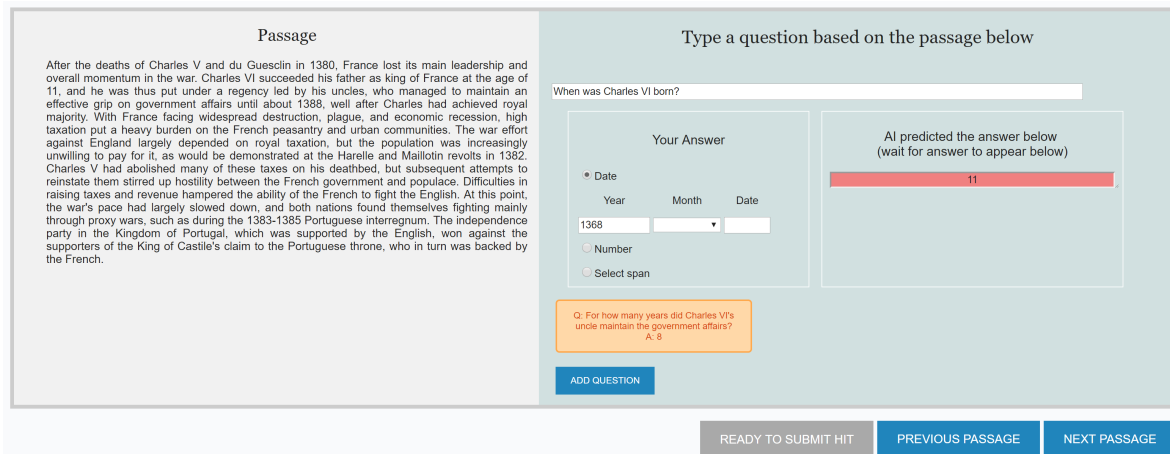


Figure 9.9: HIT interface used for collection annotations

models using *side information* or *domain knowledge* beyond labels [256, 257, 258, 259]. Especially relevant is work on supervising models using explanations [260], which, similar to our annotations, identify parts of the input that are important for prediction [261, 262]

9.4 Conclusion

We show that intermediate annotations are a cost-effective way to not only boost model performance but also alleviate certain unanticipated biases introduced during the dataset collection. However, it may be unnecessary to collect these for entire dataset and there is a sweet-spot that works best depending on the task. We proposed a simple semi-supervision technique to expose the model to these annotations. We believe that in future they can be used more directly to yield better performance gains. We have also released these annotations for the research community at https://github.com/dDua/Intermediate_Annotations.

Chapter 10

Conclusion and Future Directions

Data is the new oil.

Clive Humby

(British Mathematician)

This dissertation was motivated by the need to build better neural reasoning systems. Our results demonstrate that data is an indispensable commodity for learning neural systems, emphasizing the necessity for its meticulous curation. Natural language is contextual, ambiguous, unstructured, and compositional, making it difficult to operationalize akin to programming languages. The power of higher-order correlations through parameter and data scaling has led to the development of strong statistical models. With this work, we have initiated the exploration of new avenues for pushing neural models to engage in reasoning, decision-making and effectively assisting humans in a manner that is consistent, interpretable, and generalizable.

We show that the inadequacy of data often impedes neural systems from fully understanding how to learn and solve a specific reasoning type. Recent works in neural network scaling have shown emergent abilities [263] that are able to generalize to unseen tasks and reasoning. However, the conditions under which such abilities emerge remain unclear and mostly random. Data interventions,

on the other hand, can explicitly learn any reasoning and higher-order association without the need to rely on parameter scaling.

Limitations and Future Directions

One of the major drawbacks of data intervention schemes is the retrospective cost. It is often difficult to curate datasets for new tasks which are complete and unbiased. Multi-task training can promote positive transfer [264] across different tasks, thereby supplementing the need for complete data. However, it is difficult to know upfront which task will be similar to the new task at hand. Under the assumption that we are able to intervene with the right type of data, we outline the limitations and future prospects for the research presented in this dissertation.

- **Instance bundles:** Using bundles of related or minimally different instances has inspired several works [265, 266, 267, 268] to develop methods for curating instance bundles and imposing inference time constraints to improve textual entailment, event understanding, and commonsense reasoning. However, curating minimally different instances automatically is task and reasoning dependent, and difficult to generalize any task. It is probably better to consider such perturbations upfront while data collection.
- **Generative Passage Selection:** Generating questions for every pair of contexts can be very expensive, especially when dealing with a large document corpus. While discriminative models tend to exploit bias in data, they are less expensive at test time. Recent advances in alignment research develop preference models to reward model behavior. These reward models can learn to imitate question generation likelihood, thereby distilling the associations learned by the generative model into a discriminative model.
- **Successive Prompting:** Previously, neural approaches employed non-modular, end-to-end differentiable architectures due to the inherent challenges in seamlessly integrating discrete

outputs from external APIs into a continuous framework. However, LLMs have led to a resurgence in the modularization of tasks. This is because these models are able to (1) learn to decompose in a gradient-free or in-context [269, 270] way without the need for training with continuous variables, (2) code-switch, being pre-trained on StackOverflow style forums, making it easier to transition between symbolic and natural language surface forms. However, the current task modularization approaches predominantly follow a top-down decomposition which is effective when the decomposition plan for a similar example is available for supervision. To achieve compositional generalization, the concept of modularity of mind [271] in cognitive sciences suggests that a combination of top-down plan generation and a bottom-up exploration for module composition can help us reach intelligence. Investigating training strategies and loss functions that are able to do a global top-down plan generation and a bottom-up local module execution and composition can aid in the emergence of abilities that are able to reason about unseen compositional tasks.

Bibliography

- [1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [2] Yichen Jiang and Mohit Bansal. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop qa. *arXiv preprint arXiv:1906.07132*, 2019.
- [3] Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *AAAI*, pages 9073–9080, 2020.
- [4] Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. Hierarchical graph network for multi-hop question answering. *arXiv preprint arXiv:1911.03631*, 2019.
- [5] Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. Unsupervised question decomposition for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880, Online, 2020. Association for Computational Linguistics.
- [6] Dheeru Dua, Emma Strubell, Sameer Singh, and Pat Verga. To adapt or to annotate: Challenges and interventions for domain adaptation in open-domain question answering. *arXiv preprint arXiv:2212.10381*, 2022.
- [7] Dheeru Dua, Shivanshu Gupta, Sameer Singh, and Matt Gardner. Successive prompting for decomposing complex questions. *arXiv preprint arXiv:2212.04092*, 2022.
- [8] Dheeru Dua, Shruti Bhosale, Vedanuj Goswami, James Cross, Mike Lewis, and Angela Fan. Tricks for training sparse translation models. *arXiv preprint arXiv:2110.08246*, 2021.
- [9] Dheeru Dua, Pradeep Dasigi, Sameer Singh, and Matt Gardner. Learning with instance bundles for reading comprehension. *ArXiv preprint*, abs/2104.08735, 2021.
- [10] Dheeru Dua, Cicero Nogueira dos Santos, Patrick Ng, Ben Athiwaratkun, Bing Xiang, Matt Gardner, and Sameer Singh. Generative context pair selection for multi-hop question answering. *arXiv preprint arXiv:2104.08744*, 2021.
- [11] Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. Evaluating models’

- local decision boundaries via contrast sets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1307–1323, 2020.
- [12] Qiang Ning, Hao Wu, Pradeep Dasigi, Dheeru Dua, Matt Gardner, Robert L Logan IV, Ana Marasovic, and Zhen Nie. Easy, reproducible and quality-controlled data collection with crowdqa. *arXiv preprint arXiv:2010.06694*, 2020.
- [13] Dheeru Dua, Sameer Singh, and Matt Gardner. Benefits of intermediate annotations in reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5627–5634, 2020.
- [14] Ananth Gottumukkala, Dheeru Dua, Sameer Singh, and Matt Gardner. Dynamic sampling strategies for multi-task reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 920–924, Online, 2020. Association for Computational Linguistics.
- [15] Dheeru Dua, Ananth Gottumukkala, Alon Talmor, Sameer Singh, and Matt Gardner. Orb: An open reading benchmark for comprehensive evaluation of machine reading comprehension. In *Proceedings of the Second Workshop on Machine Reading for Question Answering*, pages 147–153, 2019.
- [16] Jun Seok Kang, Robert L Logan IV, Zewei Chu, Yang Chen, Dheeru Dua, Kevin Gimpel, Sameer Singh, and Niranjan Balasubramanian. Pomo: Generating entity-specific post-modifiers in context. *arXiv preprint arXiv:1904.03111*, 2019.
- [17] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.
- [18] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*, 2017.
- [19] Edward L Thorndike. Reading as reasoning: A study of mistakes in paragraph reading. *Reading Research Quarterly*, 6(4):425–434, 1971.
- [20] Wendy G Lehnert. A conceptual theory of question answering. In *Proceedings of the 5th international joint conference on Artificial intelligence-Volume 1*, pages 158–164, 1977.
- [21] Allen Newell and Herbert A. Simon. The logic theory machine—a complex information processing system. *IRE Trans. Inf. Theory*, 2:61–79, 1956.
- [22] Wendy Grace Lehnert. *The Process of Question Answering*. Yale University, 1977.
- [23] Charles J Fillmore. Scenes-and-frames semantics. 1977.

- [24] Karin Kipper Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania, 2005.
- [25] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.
- [26] Christiane Fellbaum. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer, 2010.
- [27] Nick Siegel, Keith Goolsbey, Robert Kahlert, and Gavin Matthews. The cyc system: Notes on architecture. *Cycorp, Inc*, 9, 2004.
- [28] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035, 2019.
- [29] Stan Franklin. Autonomous agents as embodied ai. *Cybernetics & Systems*, 28(6):499–520, 1997.
- [30] John R Searle. Minds, brains, and programs. *Behavioral and brain sciences*, 3(3):417–424, 1980.
- [31] A Simon Herbert. *The sciences of the artificial (1969)*, 1996.
- [32] J. Devlin, M. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Association for Computational Linguistics (NAACL)*, 2019.
- [33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.
- [34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [35] Yu-An Chung, Chenguang Zhu, and Michael Zeng. Splat: Speech-language joint pre-training for spoken language understanding. *NAACL*, 2020.
- [36] Edward Feigenbaum, Pamela McCorduck, and H Penny Nii. *The rise of the expert company*. Times Books, 1988.
- [37] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanus Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

- [38] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.
- [39] Brenden M Lake. Compositional generalization through meta sequence-to-sequence learning. *Advances in neural information processing systems*, 32, 2019.
- [40] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, 2018. Association for Computational Linguistics.
- [41] Michael RW Dawson. *Mind, body, world: foundations of cognitive science*. Athabasca University Press, 2013.
- [42] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- [43] Raymond J Mooney. Learning for semantic parsing. In *Computational Linguistics and Intelligent Text Processing: 8th International Conference, CICLing 2007, Mexico City, Mexico, February 18-24, 2007. Proceedings 8*, pages 311–324. Springer, 2007.
- [44] Percy Liang. Lambda dependency-based compositional semantics. *arXiv preprint arXiv:1309.4408*, 2013.
- [45] Rafael Glauber and Daniela Barreiro Claro. A systematic mapping study on open information extraction. *Expert Systems with Applications*, 112:372–387, 2018.
- [46] Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? *arXiv preprint arXiv:2010.12725*, 2020.
- [47] Roger C Schank and Robert P Abelson. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology press, 2013.
- [48] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.
- [49] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *ICLR*, 2017.

- [50] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *ICLR*, 2018.
- [51] P. Rajpurkar, R. Jia, and P. Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*, 2018.
- [52] Yann LeCun, Larry Jackel, Leon Bottou, A Brunot, Corinna Cortes, John Denker, Harris Drucker, Isabelle Guyon, UA Muller, Eduard Sackinger, et al. Comparison of learning algorithms for handwritten digit recognition. In *International conference on artificial neural networks*, volume 60, pages 53–60. Perth, Australia, 1995.
- [53] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.
- [54] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [55] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [56] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [57] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- [58] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.
- [59] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2021.
- [60] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [61] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE, 2012.
- [62] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.

- [63] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [64] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [65] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [66] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.
- [67] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr, 2013.
- [68] Kwangyoum Kim, Felix Wu, Yifan Peng, Jing Pan, Prashant Sridhar, Kyu J Han, and Shinji Watanabe. E-branchformer: Branchformer with enhanced merging for speech recognition. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 84–91. IEEE, 2023.
- [69] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.
- [70] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [71] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. *arXiv preprint arXiv:2305.05665*, 2023.
- [72] Wilson L Taylor. “cloze procedure”: A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953.
- [73] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, 2016.
- [74] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online, 2020. Association for Computational Linguistics.

- [75] Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *NAACL-HLT*, 2018.
- [76] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2016.
- [77] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.
- [78] T. Kočiský, J. Schwarz, P. Blunsom, C. Dyer, K. Hermann, G. Melis, and E. Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics (TACL)*, 6:317—328, 2018.
- [79] Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. Massively multi-lingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*, 2019.
- [80] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, 2022.
- [81] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.
- [82] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.
- [83] Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*, 2019.
- [84] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.
- [85] Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. Textworld: A learning environment for text-based games. In *Computer Games: 7th Workshop, CGW 2018*,

Held in Conjunction with the 27th International Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, July 13, 2018, Revised Selected Papers 7, pages 41–75. Springer, 2019.

- [86] Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. Torque: A reading comprehension dataset of temporal ordering questions. *arXiv preprint arXiv:2005.00242*, 2020.
- [87] Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjmeshidi. Spartqa: A textual question answering benchmark for spatial reasoning. *arXiv preprint arXiv:2104.05832*, 2021.
- [88] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [89] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017.
- [90] Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van den Broeck. On the paradox of learning to reason from data. *arXiv preprint arXiv:2205.11502*, 2022.
- [91] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *Proc. of NAACL*, 2018.
- [92] Matt Gardner, William Merrill, Jesse Dodge, Matthew E Peters, Alexis Ross, Sameer Singh, and Noah A Smith. Competency problems: On finding and removing artifacts in language data. *arXiv preprint arXiv:2104.08646*, 2021.
- [93] Robert C Moore. Problems in logical form. In *19th Annual Meeting of the Association for Computational Linguistics*, pages 117–124, 1981.
- [94] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew G Broadhead, and Oren Etzioni. Open information extraction from the web. In *IJCAI*, 2007.
- [95] Ori Yoran, Alon Talmor, and Jonathan Berant. Turning tables: Generating examples from semi-structured tables for endowing language models with reasoning skills. *ArXiv preprint, abs/2107.07261*, 2021.
- [96] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.
- [97] OpenAI. GPT-4: A language model for natural language processing. OpenAI Blog, 2023.
- [98] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv preprint, abs/2201.11903*, 2022.

- [99] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.
- [100] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.
- [101] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.
- [102] Akari Asai and Hannaneh Hajishirzi. Logic-guided data augmentation and regularization for consistent question answering. *arXiv preprint arXiv:2004.10157*, 2020.
- [103] Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. Reasoning over paragraph effects in situations. *arXiv preprint arXiv:1908.05852*, 2019.
- [104] Noah A Smith and Jason Eisner. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pages 354–362, 2005.
- [105] Sean Welleck, Ilya Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*, 2019.
- [106] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.
- [107] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.
- [108] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.
- [109] Jacob Andreas. Good-enough compositional data augmentation. *arXiv preprint arXiv:1904.09545*, 2019.
- [110] Ran Zmigrod, Sabrina J Mielke, Hanna Wallach, and Ryan Cotterell. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. *arXiv preprint arXiv:1906.04571*, 2019.
- [111] Mike Lewis and Angela Fan. Generative question answering: Learning to answer the whole question. In *International Conference on Learning Representations*, 2018.
- [112] Jiangming Liu and Matt Gardner. Multi-step inference for reasoning over paragraphs. *arXiv preprint arXiv:2004.02995*, 2020.
- [113] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

- [114] Yuntian Deng and Alexander M Rush. Cascaded text generation with markov transformers. *arXiv preprint arXiv:2006.01112*, 2020.
- [115] Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. An empirical investigation of global and local normalization for recurrent neural sequence models using a continuous relaxation to beam search. *arXiv preprint arXiv:1904.06834*, 2019.
- [116] Pradeep Dasigi, Nelson F Liu, Ana Marasovic, Noah A Smith, and Matt Gardner. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.
- [117] Hengyi Cai, Hongshen Chen, Yonghao Song, Zhuoye Ding, Yongjun Bao, Weipeng Yan, and Xiaofang Zhao. Group-wise contrastive learning for neural dialogue generation. *arXiv preprint arXiv:2009.07543*, 2020.
- [118] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013.
- [119] Hiroshi Noji and Hiroya Takamura. An analysis of the utility of explicit negative examples to improve the syntactic abilities of neural language models. *arXiv preprint arXiv:2004.02451*, 2020.
- [120] Bo Dai and Dahua Lin. Contrastive learning for image captioning. *arXiv preprint arXiv:1710.02534*, 2017.
- [121] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.
- [122] Yi-Ting Yeh and Yun-Nung Chen. Qainfomax: Learning robust question answering system by mutual information maximization. *arXiv preprint arXiv:1909.00215*, 2019.
- [123] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.
- [124] Divyansh Kaushik, Amrith Setlur, Eduard Hovy, and Zachary C Lipton. Explaining the efficacy of counterfactually-augmented data. *arXiv preprint arXiv:2010.02114*, 2020.
- [125] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*, 2020.
- [126] Alexis Ross, Ana Marasović, and Matthew E. Peters. Explaining nlp models via minimal contrastive editing (mice). *ArXiv*, abs/2012.13985, 2020.
- [127] Tongshuang Wu, Marco Túlio Ribeiro, J. Heer, and Daniel S. Weld. Polyjuice: Automated, general-purpose counterfactual generation. *ArXiv*, abs/2101.00288, 2021.

- [128] Kai-Wei Chang, Rajhans Samdani, Alla Rozovskaya, Nick Rizzolo, Mark Sammons, and Dan Roth. Inference protocols for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 40–44, 2011.
- [129] Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840*, 2019.
- [130] Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *arXiv preprint arXiv:2102.01017*, 2021.
- [131] Pasquale Minervini and Sebastian Riedel. Adversarially regularising neural nli models to integrate logical background knowledge. In *CoNLL*, 2018.
- [132] Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. A logic-driven framework for consistency of neural models. *arXiv preprint arXiv:1909.00126*, 2019.
- [133] Damien Teney, Ehsan Abbasnejad, and Anton van den Hengel. On incorporating semantic prior knowledge in deep learning through embedding-space constraints. *arXiv preprint arXiv:1909.13471*, 2019.
- [134] Damien Teney, Ehsan Abbasnejad, and Anton van den Hengel. Learning what makes a difference from counterfactual examples and gradient supervision. *arXiv preprint arXiv:2004.09034*, 2020.
- [135] Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. Contrastive explanations for model interpretability. *arXiv preprint arXiv:2103.01378*, 2021.
- [136] Nitish Gupta, S. Singh, Matt Gardner, and D. Roth. Paired examples as indirect supervision in latent decision models. *ArXiv*, abs/2104.01759, 2021.
- [137] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302, 2018.
- [138] Mor Geva, Yoav Goldberg, and Jonathan Berant. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.
- [139] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *NAACL*, 2018.
- [140] Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. Compositional questions do not necessitate multi-hop reasoning. *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

- [141] Jifan Chen, Shih-ting Lin, and Greg Durrett. Multi-hop question answering via reasoning chains. *arXiv preprint arXiv:1910.02610*, 2019.
- [142] Nan Shao, Yiming Cui, Ting Liu, Shijin Wang, and Guoping Hu. Is graph structure necessary for multi-hop reasoning? *arXiv preprint arXiv:2004.03096*, 2020.
- [143] Shai Ben-David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, pages 129–136, 2010.
- [144] Christopher Clark and Matt Gardner. Simple and effective multi-paragraph reading comprehension. *arXiv preprint arXiv:1710.10723*, 2017.
- [145] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. Multi-step retriever-reader interaction for scalable open-domain question answering. *arXiv preprint arXiv:1905.05733*, 2019.
- [146] Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*, 2017.
- [147] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.
- [148] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.
- [149] Minjoon Seo, Tom Kwiatkowski, Ankur P Parikh, Ali Farhadi, and Hannaneh Hajishirzi. Phrase-indexed question answering: A new challenge for scalable document comprehension. *arXiv preprint arXiv:1804.07726*, 2018.
- [150] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.
- [151] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*, 2020.
- [152] Cicero dos Santos, Xiaofei Ma, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. Beyond [cls] through ranking by generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1722–1727, 2020.
- [153] Danqi Chen, Jason Bolton, and Christopher D. Manning. A thorough examination of the cnn/daily mail reading comprehension task. 2016.

- [154] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*. Association for Computational Linguistics, 2017.
- [155] Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI*, 2005.
- [156] Pengcheng Yin and Graham Neubig. A syntactic neural model for general-purpose code generation. In *ACL'17*, 2017.
- [157] David L Chen and Raymond J Mooney. Learning to interpret natural language navigation instructions from observations. In *AAAI*, volume 2, pages 1–2, 2011.
- [158] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, 2013.
- [159] Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. Neural semantic parsing with type constraints for semi-structured tables. In *EMNLP*, 2017.
- [160] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *ACL*, 2015.
- [161] Marie-Catherine de Marneffe and Christopher D. Manning. The stanford typed dependencies representation. In *CFCFPE@COLING*, 2008.
- [162] Martha Palmer, Daniel Gildea, and Nianwen Xue. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103, 2010.
- [163] Timothy Dozat, Peng Qi, and Christopher D. Manning. Stanford’s graph-based neural dependency parser at the conll 2017 shared task. In *CoNLL Shared Task*, 2017.
- [164] Luheng He, Kenton Lee, Mike Lewis, and Luke S. Zettlemoyer. Deep semantic role labeling: What works and what’s next. In *ACL*, 2017.
- [165] Gabriel Stanovsky, Julian Michael, Luke S. Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *NAACL-HLT*, 2018.
- [166] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.
- [167] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL-HLT*, 2018.
- [168] Christopher Clark and Matt Gardner. Simple and effective multi-paragraph reading comprehension. In *ACL*, 2018.
- [169] Divyansh Kaushik and Zachary Chase Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *EMNLP*, 2018.

- [170] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, 2016.
- [171] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, 2015.
- [172] Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *ArXiv preprint*, abs/2112.00114, 2021.
- [173] Ehud Karpas, Omri Abend, Yonatan Belinkov, Barak Lenz, Opher Lieber, Nir Ratner, Yoav Shoham, Hofit Bata, Yoav Levine, Kevin Leyton-Brown, et al. Mrkl systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. *ArXiv preprint*, abs/2205.00445, 2022.
- [174] Tushar Khot, Daniel Khashabi, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Text modular networks: Learning to decompose tasks in the language of existing models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1264–1279, Online, 2021. Association for Computational Linguistics.
- [175] Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, Dublin, Ireland, 2022. Association for Computational Linguistics.
- [176] Elad Segal, Avia Efrat, Mor Shoham, Amir Globerson, and Jonathan Berant. A simple and effective model for answering multi-span questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3074–3080, Online, 2020. Association for Computational Linguistics.
- [177] Yue Jin, Tianqing Zheng, Chao Gao, and Guoqiang Xu. Mtmsn: Multi-task and multi-modal sequence network for facial action unit and expression recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3597–3602, 2021.
- [178] Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. Neural module networks for reasoning over text. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.
- [179] Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V. Le. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.
- [180] Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. Giving BERT a calculator: Finding operations and arguments with reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International*

- Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5947–5952, Hong Kong, China, 2019. Association for Computational Linguistics.
- [181] Xiaofei Ma, Cicero Nogueira dos Santos, and Andrew O. Arnold. Contrastive fine-tuning improves robustness for neural rankers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 570–582, Online, 2021. Association for Computational Linguistics.
- [182] Dheeraj Rajagopal, Siamak Shakeri, Cicero Nogueira dos Santos, Eduard Hovy, and Chung-Ching Chang. Counterfactual data augmentation improves factuality of abstractive summarization. *ArXiv preprint*, abs/2205.12416, 2022.
- [183] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics.
- [184] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don’t just assume; look and answer: Overcoming priors for visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4971–4980. IEEE Computer Society, 2018.
- [185] Timo Schick. *Few-shot learning with language models: Learning from instructions and contexts*. PhD thesis, lmu, 2022.
- [186] Ana Marasović, Iz Beltagy, Doug Downey, and Matthew E Peters. Few-shot self-rationalization with natural language prompts. *ArXiv preprint*, abs/2111.08284, 2021.
- [187] Eric Zelikman, Yuhuai Wu, and Noah D Goodman. Star: Bootstrapping reasoning with reasoning. *ArXiv preprint*, abs/2203.14465, 2022.
- [188] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models. *ArXiv preprint*, abs/2205.10625, 2022.
- [189] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. *ArXiv preprint*, abs/2210.03350, 2022.
- [190] Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. *ArXiv preprint*, abs/2210.02406, 2022.
- [191] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True few-shot learning with language models. *Advances in Neural Information Processing Systems*, 34, 2021.
- [192] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. *ArXiv preprint*, abs/2112.08633, 2021.

- [193] Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. A multi-type multi-span network for reading comprehension that requires discrete reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1596–1606, Hong Kong, China, 2019. Association for Computational Linguistics.
- [194] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.
- [195] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*, 2019.
- [196] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 2022.
- [197] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.
- [198] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*, 2020.
- [199] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*, 2021.
- [200] Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. *arXiv preprint arXiv:1910.09753*, 2019.
- [201] Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*, 2017.
- [202] Georgios Balikas, Anastasia Krithara, Ioannis Partalas, and George Paliouras. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In *International Workshop on Multimodal Retrieval in the Medical Domain*, pages 26–39. Springer, 2015.
- [203] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*, 2016.
- [204] Simon Šuster and Walter Daelemans. Clicr: a dataset of clinical case reports for machine reading comprehension. 2018.
- [205] Juliano Rabelo, Randy Goebel, Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021. *The Review of Socionetwork Strategies*, 16(1):111–133, 2022.

- [206] Stephen E Robertson and Karen Spärck Jones. Simple, proven approaches to text retrieval. Technical report, University of Cambridge, Computer Laboratory, 1994.
- [207] Ori Ram, Gal Shachaf, Omer Levy, Jonathan Berant, and Amir Globerson. Learning to retrieve passages without supervision. *arXiv preprint arXiv:2112.07708*, 2021.
- [208] Gerhard Widmer and Miroslav Kubát. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23:69–101, 2004.
- [209] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, page 114, 2004.
- [210] Leonid V Kantorovich. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422, 1960.
- [211] Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. Simple entity-centric questions challenge dense retrievers. *arXiv preprint arXiv:2109.08535*, 2021.
- [212] Zhijing Jin, Julius von Kügelgen, Jingwei Ni, Tejas Vaidhya, Ayush Kaushal, Mrinmaya Sachan, and Bernhard Schölkopf. Causal direction of data collection matters: Implications of causal and anticausal learning for nlp. *arXiv preprint arXiv:2110.03618*, 2021.
- [213] Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Yoshua Bengio, and Adam Trischler. Neural models for key phrase detection and question generation. *arXiv preprint arXiv:1706.04560*, 2017.
- [214] Siamak Shakeri, Cicero Nogueira dos Santos, Henry Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. End-to-end synthetic data generation for domain adaptation of question answering systems. *arXiv preprint arXiv:2010.06028*, 2020.
- [215] Kalpesh Krishna and Mohit Iyyer. Generating question-answer hierarchies. *arXiv preprint arXiv:1906.02622*, 2019.
- [216] Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. Leveraging context information for natural question generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 569–574, 2018.
- [217] Tassilo Klein and Moin Nabi. Learning to answer by learning to ask: Getting the best of gpt-2 and bert worlds. *arXiv preprint arXiv:1911.02365*, 2019.
- [218] Tanya Goyal, Junyi Jessy Li, and Greg Durrett. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*, 2022.
- [219] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*, 2020.

- [220] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544, 2013.
- [221] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*, 2022.
- [222] Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. *arXiv preprint arXiv:2004.14503*, 2020.
- [223] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *arXiv preprint arXiv:2210.02627*, 2022.
- [224] Hangfeng He, Mingyuan Zhang, Qiang Ning, and Dan Roth. Foreseeing the benefits of incidental supervision. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1782–1800, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics.
- [225] Dheeru Dua, Shruti Bhosale, Vedanuj Goswami, James Cross, Mike Lewis, and Angela Fan. Tricks for training sparse translation models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3340–3345, Seattle, United States, 2022. Association for Computational Linguistics.
- [226] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1):5232–5270, 2022.
- [227] Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea, 2012. Association for Computational Linguistics.
- [228] Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. Reasoning over paragraph effects in situations. *arXiv preprint arXiv:1908.05852*, 2019.
- [229] Alon Talmor and Jonathan Berant. Multiqa: An empirical investigation of generalization and transfer in reading comprehension. *57th Annual Conference of the Association for Computational Linguistics (ACL)*, abs/1905.13453, 2019.
- [230] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, 2018.

- [231] Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 6174–6184, 2019.
- [232] Haichao Zhu, Li Dong, Furu Wei, Wenhui Wang, Bing Qin, and Ting Liu. Learning to ask unanswerable questions for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4238–4248, Florence, Italy, July 2019. Association for Computational Linguistics.
- [233] A. Saha, R. Aralikkatte, M. Khapra, and K. Sankaranarayanan. Duorc: Towards complex language understanding with paraphrased reading comprehension. In *Association for Computational Linguistics (ACL)*, 2018.
- [234] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman. NewsQA: A machine comprehension dataset. In *Workshop on Representation Learning for NLP*, 2017.
- [235] Asa Cooper Stickland and Iain Murray. BERT and PALs: Projected attention layers for efficient adaptation in multi-task learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5986–5995, Long Beach, California, USA, 09–15 Jun 2019. PMLR.
- [236] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics.
- [237] A. Conneau and D. Kiela. Senteval: An evaluation toolkit for universal sentence representations. *Association for Computational Linguistics (ACL)*, 2018.
- [238] A. H. Miller, W. Feng, A. Fisch, J. Lu, D. Batra, A. Bordes, D. Parikh, and J. Weston. Parlai: A dialog research software platform. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–84, 2017.
- [239] J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards AI-complete question answering: A set of prerequisite toy tasks. *ICLR*, 2016.
- [240] Alon Talmor and Jonathan Berant. Multiqa: An empirical investigation of generalization and transfer in reading comprehension. *Association for Computational Linguistics*, 2019.
- [241] Yichong Xu, Xiaodong Liu, Yelong Shen, Jingjing Liu, and Jianfeng Gao. Multi-task learning with sample re-weighting for machine reading comprehension. *North American Chapter of the Association for Computational Linguistics*, 2019.
- [242] Gail A Carpenter and Stephen Grossberg. The art of adaptive pattern recognition by a self-organizing neural network. *Computer*, 21(3):77–88, 1988.

- [243] Victor Sanh, Thomas Wolf, and Sebastian Ruder. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6949–6956, 2019.
- [244] Sahil Sharma and Balaraman Ravindran. Online multi-task learning using active sampling. 2017.
- [245] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. 2017.
- [246] T. Kočiský, J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, and E. Grefenstette. The NarrativeQA reading comprehension challenge. *arXiv preprint arXiv:1712.07040*, 2017.
- [247] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL*, 2019.
- [248] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *NeurIPS*, 2019.
- [249] Varun Manjunatha, Nirat Saini, and Larry S Davis. Explicit bias discovery in visual question answering models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9562–9571, 2019.
- [250] Pierre Machart and Liva Ralaivola. Confusion matrix stability bounds for multiclass classification. *arXiv preprint arXiv:1202.6221*, 2012.
- [251] Sokol Koço and Cécile Capponi. On multi-class classification through the minimization of the confusion matrix norm. In *Asian Conference on Machine Learning*, pages 277–292, 2013.
- [252] Omar Zaidan, Jason Eisner, and Christine Piatko. Using “annotator rationales” to improve machine learning for text categorization. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*, pages 260–267, 2007.
- [253] Jeff Donahue and Kristen Grauman. Annotator rationales for visual recognition. In *2011 International Conference on Computer Vision*, pages 1395–1402. IEEE, 2011.
- [254] Siva Reddy, Danqi Chen, and Christopher D Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.
- [255] Qiang Ning, Hangfeng He, Chuchu Fan, and Dan Roth. Partial or complete, that’s the question. *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.
- [256] Gideon S. Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 870–878, 2008.

- [257] Ming-Wei Chang, Lev Ratinov, and Dan Roth. Guiding semi-supervision with constraint-driven learning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 280–287, 2007.
- [258] Kuzman Ganchev, Joao Graca, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research (JMLR)*, July 2010.
- [259] Tim Rocktaschel, Sameer Singh, and Sebastian Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015.
- [260] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *IJCAI*, 2017.
- [261] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.
- [262] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.
- [263] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.
- [264] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.
- [265] Eric Mitchell, Joseph J Noh, Siyan Li, William S Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher D Manning. Enhancing self-consistency and performance of pre-trained language models through natural language inference. *arXiv preprint arXiv:2211.11875*, 2022.
- [266] Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. Maieutic prompting: Logically consistent reasoning with recursive explanations. *arXiv preprint arXiv:2205.11822*, 2022.
- [267] Mor Geva, Tomer Wolfson, and Jonathan Berant. Break, perturb, build: Automatic perturbation of reasoning paths through question decomposition. *Transactions of the Association for Computational Linguistics*, 10:111–126, 2022.
- [268] Junru Lu, Xingwei Tan, Gabriele Pergola, Lin Gui, and Yulan He. Event-centric question answering via contrastive learning and invertible event transformation. *arXiv preprint arXiv:2210.12902*, 2022.
- [269] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023.

[270] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

[271] Jerry A Fodor. *The modularity of mind*. MIT press, 1983.