Adversarial Robustness of Language Models with Humans and Models in the Loop

Max Bartolo

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Computer Science

University College London

July 11, 2024

To my family and friends, whose love, support and encouragement made this journey possible.

Acknowledgements

I would like to express my sincere gratitude to all those who supported me throughout this research journey, without whom this work would not have been possible.

Firstly, I extend my deepest appreciation to my supervisor, Pontus Lars Erik Saito Stenetorp, for his invaluable guidance, insightful feedback, and unwavering support during my research journey. Pontus is incredibly thoughtful and kind, and one of the most authentic people I know. I feel incredibly lucky to have been his first PhD student, and will deeply miss our conversations, both about research and life in general. His expertise and mentorship have been instrumental in shaping this work, and my personal development as a researcher. I would also like to thank Sebastian Riedel, for inspiring me to pursue research in Natural Language Processing from the very first lecture at UCL, for sharing his wisdom as my secondary supervisor, for demonstrating that it was possible to find model-fooling questions during early prototyping with the "Beat the AI" interface, and for the past eight years of mentorship and guidance.

I am grateful to have crossed paths with so many remarkable individuals. Our collaborations, discussions, and deep conversations over the past few years will remain cherished memories. I would like to express my appreciation for all my co-authors, collaborators, colleagues, and friends at University College London (UCL), Facebook AI Research (FAIR), DeepMind, MLCommons, Cohere, and beyond. I cannot possibly mention everyone individually, but know that I value each and every one of your contributions. I'd like to express special thanks to my colleagues at the UCL research group for their stimulating discussions, helpful critiques, and camaraderie, including Patrick Lewis, Yuxiang Wu, Pasquale Minervini, Johannes Welbl,

Alastair Roberts, Maximilian Mozes, Tim Rocktäschel, Matko Bošnjak, Saku Sugawara, Yihong Chen, Linqing Liu, Yao Lu, Luca Franceschi, Oana-Maria Camburu, David Adelani and Xuanli He.

To the people who encouraged me to develop the NLP module at UCL's School of Management, Alastair Moore and David Alderton, and to all the students across the years for your enthusiasm, curiosity and challenging questions pushing me to deepen my understanding and sharpen my intuition around core concepts, thank you. I would also like to thank Niall Roche, Kamil Tylinski, Jiangbo Shangguan, Walter Hernandez and Daniel Hoadley.

Further thanks goes to Douwe Kiela and Robin Jia for their mentorship and supervision at FAIR, along with my collaborators and colleagues including Adina Williams, Tristan Thrush, Hannah Rose Kirk, Pedro Rodriguez, Katerina Margatina, Rebecca Qian and Candace Ross. I would also like to express my thanks to all members of the DeepMind Robust and Verified AI (RVAI) team, it was a privilege working with you, with particular thanks to Johannes Welbl and Po-Sen Huang. Your contributions have been invaluable, and I am grateful for your support.

I would also like to express my appreciation towards my colleagues at MLCommons, for supporting and continuing to believe in the Dynabench platform and the research it enables. Thank you to David Kanter, Peter Mattson, Rafael Mosquera, Juan Ciro, Lilith Bat-Leah, Praveen Paritosh, Alicia Parrish, Hannah Jessica Quaye, Charvi Rastogi, Oana Inel, Vijay Janapa Reddi and Lora Aroyo.

I would also like to thank my friends and colleagues at Cohere including Acyr Locatelli, Matthias Gallé, Nick Jakobi, Ellen Gilsenan-McMahon, Phil Blunsom, Aidan Gomez, Nick Frosst, and Ivan Zhang, along with the entire Command and Modelling teams. For the encouragement and support, thank you to Seraphina Goldfarb-Tarrant, Nora Kassner and Kate Philips-Kaiser.

My deepest gratitude goes to my family and friends — thank you to my parents for your endless love and support, and to Sam, Jade and Faye for your unique and special roles in my life. Further thanks to John, Andrew, Keith and Julian for your unwavering friendship.

To Kristina, my heartfelt appreciation for sharing this journey with me. Words cannot express the depth of my gratitude for your presence by my side. Amidst the challenges of a global pandemic and the demands of parallel PhD pursuits and work commitments, you provided never-ending joy and support. Your resilience and dedication are an inspiration.

This dissertation was written with some minimal assistance from Cohere's Command R+ model, released on the 4th April 2024 with model weights made publicly available, in accordance with UCL's guidance on using and acknowledging generative AI technologies. The development of Command R+ was directly and indirectly influenced by the work presented herein.

Declaration

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

Abstract

Machine Learning (ML) systems often fail in unexpected and unpredictable ways. They lack robustness to minor non-semantic changes to inputs, which can limit their potential for widespread application. We provide a comprehensive exploration of the involvement of humans and models in the loop to study and improve the adversarial robustness of machine language understanding.

We first investigate the use of increasingly capable models in the annotation loop to collect progressively more complex and interesting data, for both training and evaluation. We further investigate the downstream generalisation, robustness and transfer implications, demonstrating improvements across all axes of interest. Following this, we introduce Dynabench, an open-source platform to facilitate dynamic dataset creation and model benchmarking, aiming for more robust and informative dynamic benchmarks across a suite of NLP tasks. Building on this foundation, we explore Synthetic Adversarial Data Generation (SADG), making models more robust to human adversaries without requiring any additional human data collection. We also introduce Adversarial Human Evaluation (AHE), an evaluation paradigm involving humans in the loop to measure robustness to adversarial attack, with implications for performance aspects such as robustness and safety.

Finally, we introduce Generative Annotation Assistants (GAAs), generator-in-the-loop models that provide real-time suggestions that annotators can either approve, modify, or entirely reject. We demonstrate the effectiveness of GAAs through a detailed study, demonstrating significant benefits in both annotation efficiency and effectiveness, which also leads to improved downstream model performance and robustness.

We offer novel insight into the potential of human-model competition and collaboration, providing a pathway to more robust and reliable language models capable of adapting to diverse adversarial scenarios, representative of the real-world environments these models are expected to operate in.

Impact Statement

This thesis investigates the involvement of humans and models in the annotation loop, leveraging the creativity and intuition of human annotators and the efficiency and adaptability of generative models, to understand and improve the robustness of Machine Learning systems. As such system capabilities improve, it is increasingly important that they are developed with human users in mind first and foremost. We focus on the task of extractive Question Answering, noting that the methods developed are broadly applicable across wider LLM and NLP applications.

The impact of the work presented in this thesis spans multiple dimensions. It has contributed to the development of generally improved and more robust language processing technologies, enhancing their performance and reliability, and making them more accessible to society. It further improves our understanding of how humans and machines interact in competitive and collaborative settings, and explores settings that benefit both — where machine learning systems benefit from human feedback to address identified blind spots, and humans benefit from generative model assistance in ways that make them more efficient and effective. Work on Synthetic Adversarial Data Generation provides an extensible approach driving model improvements beyond what existing data can provide, and becomes particularly relevant as we approach the limits of data scaling where model training requirements surpass that which all existing human-authored data is able to provide. Work on dynamically and adversarially involving humans as part of model evaluation workflows has also influenced thinking around evaluation strategies and approaches for measuring the performance of large language models.

Furthermore, the training and evaluation datasets collected and synthetically

generated in this work provide novel, publicly available resources to the research community and facilitate future work in these directions. These resources provide valuable insights and continue to inspire the development of techniques aimed at enhancing the robustness and performance of real-world systems. The open-source, permissibly-licensed model and software contributions also continue to support research in the space of generative assistive models and dynamic adversarial data collection and benchmarking.

The contributions in this thesis have been published and presented at several international journals and conferences, including TACL, EMNLP, ACL, and NAACL. Much of the work in this thesis has also been presented at invited talks in various academic and industry settings.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural Language Processing (NLP) is a field of Computer Science concerned with the development of systems that understand and generate human language. Human intellectual progress has often been attributed to language, and machine's ability to comprehend and express language has been explored since the early days of modern computing (Jefferson, 1949; Parker and Gibson, 1979). Among these earliest contributions was Alan Turing's *imitation game*, a test of machine intelligence in which a human interrogator asks a series of questions to two respondents (or witnesses), a human and a machine, solely through natural language (and ideally typed language), in an attempt to distinguish them. The machine's aim is to reliably fool the interrogator while the human participant's aim, as originally described, is to support the interrogator — a strategy which the machine can also employ to demonstrate it's ability to *think* (Turing, 1950).

The field has since progressed from rule-based and expert systems designed to elicit understanding from the linguistic properties of text, primarily those revolving around syntax and semantics (Reynolds Jr., 1954; Chomsky, 1956, 1957; Weizenbaum, 1966) and knowledge representation (Minsky, 1969; Winograd, 1971; Lenat and Guha, 1989; Sowa et al., 1992; Miller, 1995; Lafferty et al., 2001) — to large, complex machine learning models that demonstrate the ability to understand the intent and meaning of text to varying levels of ability and generate text that may appear indistinguishable from that written by a human (Radford et al., 2019). This has happened over an impressively short period of time, going from model generations

that were often incomplete, ungrammatical, nonsensical and repetitive to engaging and high quality outputs in under 10 years.

Notable milestones along the way include the foundational work laid by early pioneers defining and formalising language tasks, building datasets and structuring evaluation methodologies along with early rule-based and data-driven systems demonstrating promising performance on these tasks (Lowerre and Reddy, 1976; Brill, 1992; Wilks, 1975; Bowman et al., 2015; Simmons, 1970; Waltz, 1978; Rajpurkar et al., 2016). Recurrent Neural Networks, first described by Little in 1974, (Gurney, 1997; Hopfield, 1982; LeCun et al., 2015) and state-capturing components such as Long Short Term Memory units (LSTMs) (Hochreiter and Schmidhuber, 1997) provided architectural innovations that were attuned to handling the sequential nature of language.

Further progress was driven by richer and substantially more expressive word representations based on co-occurrence patterns with work such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), and subsequently contextual embeddings (Dai and Le, 2015; McCann et al., 2017; Peters et al., 2018; Devlin et al., 2019) which capture the nuance of the context within which words appear as a component of individual token representations.

The development of the attention mechanism, originally designed to handle the alignment problem across source and target languages in machine translation (Bahdanau et al., 2015), paved the way for a form of representation learning based on self-attention, with similar or improved expressive power but an architectural design that was considerably more adept to parallelisation, in the transformer (Vaswani et al., 2017). The sequential nature of RNNs meant that the computation of the representation of the next token required the completion of the calculation of the current one. Transformers, on the other hand, require a predefined, fixed sequence length, across which token computations can happen independently, and thus in parallel, across multiple computation nodes. Scaling both data and model sizes has driven further performance improvements across tasks, domains and modalities (Kaplan et al., 2020; Hoffmann et al., 2022).

A commonly studied NLP domain is Machine Reading Comprehension (MRC) or simply Reading Comprehension (RC), a set of tools for probing a system's comprehension abilities (Richardson et al., 2013; Hermann et al., 2015; Nguyen et al., 2016). Interest in this area of research is in part motivated by the way humans naturally seek information — asking questions. A specific, and relatively well-constrained, task formulation for investigating Reading Comprehension is Extractive Question Answering (EQA) (Rajpurkar et al., 2016). Formally, a model f, is provided with a bitext input consisting of a passage, p (sometimes referred to as a context, c) and a question, q. The task requires the model to identify or extract the answer, a within the passage. This is often modelled as predicting the start and end indices of the selected answer within the passage.

The example shown in Figure 1.1 involves passage from the Wikipedia article on Oxygen Compounds about organic compounds, along with a set of questions and highlighted ground truth answers. For the selected question "Which of the organic compounds, in the article, contains nitrogen?", various entities such as "alcohols", "ethers", "ketones", "aldehydes", "carboxylic acids", "esters", "acid anydrides", and "amides" are described as organic compounds and thus satisfy the first criterion that the question poses. However, identifying the compound that contains Nitrogen requires both the knowledge and ability to resolve to Nitrogen's chemical element, N, and the ability to infer which of the compound compositions contain (and which don't) the letter "N". The ground truth answer, "amides", is highlighted in the passage. This particular example demonstrates the type of *multi-hop reasoning* capabilities a system would require to arrive at the correct answer.

Question Answering provides a particularly interesting test bed for researching the intricacies and effects of various system capabilities (Ferrucci et al., 2010) and has also been used as a reference task for exploring models that generalise to many different kinds of NLP tasks (McCann et al., 2019).

A broad categorisation of capabilities that Reading Comprehension can help understand is *reasoning*. This covers a diverse set of complex comprehension skills, and can take on various interpretations. At a high level, reasoning abilities can

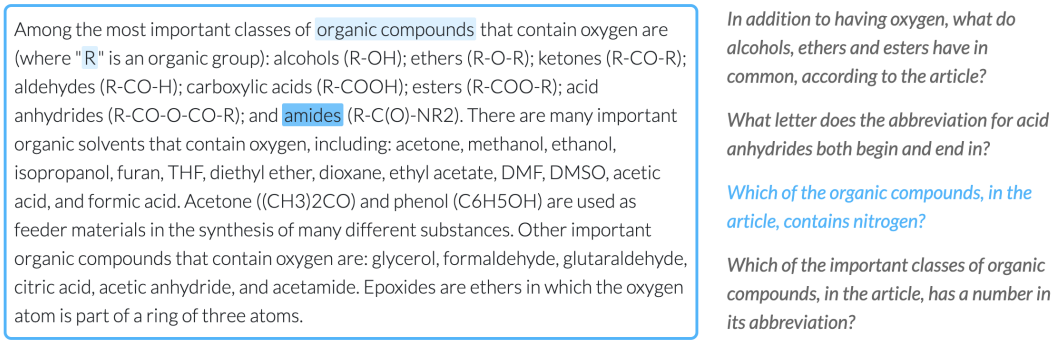


Figure 1.1: A set of examples from an Extractive Question Answering (EQA) dataset collected as part of this work showing a passage, a set of questions and corresponding answer spans in the passage.

be demonstrated through a variety of skills, such as inferring a conclusion from premises, planning, identifying optimal solutions, or resolving logical primitives, yet it remains challenging to comprehensively define (Wason and Johnson-Laird, 1972; Manktelow, 1999).

We provide a more detailed breakdown in Chapter 3, but as a quick introduction, some of the main capabilities investigated include: paraphrasing, the abilities to process external knowledge and resolve co-references, handling negation, as well as comparative, numeric, temporal, spatial and inductive reasoning. It is interesting to note that some of the earliest work to systematically investigate reasoning in the cognitive psychology literature focuses on *negation* (Wason, 1959, 1961) — a capability that the work presented in this chapter demonstrates considerable improvements on.

Reasoning skills are also closely interlinked with *robustness* — the ability of systems to consistently predict the correct outcome given non-semantic differences across inputs or input types. It stands to reason that if a system has *truly* gained a reasoning ability, that is it has effectively learnt a general function that reflects the skill or capability that allows it to perform a particular task, it must then do so robustly. If it does not, and for example fails in ways that contradict a set of logical rules that define the capability, can that system be said to be truly capable of reasoning? To some extent, any exhibition of performance beyond expectation on a task that strictly *requires* reasoning abilities can be interpreted as a

system having accomplished some level of reasoning, although it is often challenging to disambiguate what demonstrations require true reasoning abilities, and what can be achieved through other means, such as by exploiting spurious data correlations (Weissenborn et al., 2017). We posit that *true reasoning* requires consistent and robust demonstration of the system's abilities to perform a given reasoning task, and further requires that failures occur in relatively expected and predictable ways.

Neural Language Models (LMs) (Bengio et al., 2000) are general systems which take text as input and generate text as output. This provides a degree of flexibility that makes them well-suited for investigating reasoning and robustness, particularly in the context of settings involving complex interactions between machines and humans. From a statistical perspective, Language Models capture the joint distribution of a sequence of words (or tokens), which by application of Bayes' theorem, be represented as the conditional probability of the next word in a sequence given all the previous words, as:

$$P(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} P(w_t \mid w_1, w_2, \ldots, w_{t-1})$$

Language Models can capture rich representations of input text which makes them well-suited for application to downstream tasks (Dai and Le, 2015; Peters et al., 2017; Howard and Ruder, 2018; Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019b; Clark et al., 2020) as well as text generation (Radford et al., 2018; Brown et al., 2020). This work makes use of and investigates a variety of Language Models, which we describe in the corresponding chapters.

1.1 Aims & Themes

Recent advances in machine reading comprehension capability, driven primarily by the adoption of the highly parallelisable transformer architecture (Vaswani et al., 2017) enabling large scale pre-training of contextual token representations (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019b), have driven machine performance to surpass humans on established benchmarks such as SQuAD1.1 (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017) and SQuAD2.0 (Rajpurkar et al., 2018), as

well as more general improvements on language understanding benchmarks such as GLUE (Wang et al., 2018).

Despite these impressive results, there still exist substantial headroom gaps (although these have decreased during the period over which this work was conducted, including through the contributions made in this work) between such reading comprehension model performance and that of humans on tasks specifically designed to pose challenging comprehension requirements — such as multi-hop (Welbl et al., 2018; Yang et al., 2018a) or comparative questions (Dua et al., 2019). This suggests that machine reading comprehension is far from "solved".

While this indicates that contemporary RC models do not possess the full spectrum of reasoning capabilities required for *true reading comprehension*, there is limited understanding of what comprehension requirements models fail to demonstrate convincingly, and why. Furthermore, the field lacks a clear direction for improving RC model reasoning capabilities.

This thesis seeks to address this knowledge gap by providing an understanding of the reasoning capabilities and failure modes of reading comprehension systems, and exploring ways to improve system robustness by involving humans and machines at various stages of the data collection and model evaluation processes.

Aims. We aim to develop an in-depth understanding of the competencies of contemporary state-of-the-art Reading Comprehension models, primarily from the perspective of their reasoning capabilities and robustness characteristics. This requires a general approach that can adapt to different model abilities and behavioural nuances, and which also allows the design of creative probing attacks and the exploitation identified failure modes. To accomplish this, we aim to explore how models and humans interact both competitively and collaboratively. Once we understand where systems can improve, we aim to develop new methodologies to enhance model comprehension capabilities and adversarial robustness, by building on this synergistic interaction between human and machine.

Themes. We consistently touch on several key themes throughout this work: i) *robustness*, or the ability of systems to more consistently behave as expected, which

we explore and seek to understand in Chapter 3, and then design strategies for improving in Chapters 5 and 6; ii) *adversarial interaction*, or asking humans to probe for failure modes in language models, for which we lay the groundwork in Chapter 3 and expand on in Chapter 4; iii) involving *humans and models in the loop*, a theme which permeates throughout this work in various forms and setups, and iv) more *reliable evaluation*, particularly focused on providing evaluative insight into characteristics that current approaches do not measure, where we introduce new adversarial human-generated test sets in Chapter 3, describe a research platform for dynamic evaluation and benchmarking in Chapter 4, introduce adversarial human evaluation in Chapter 5 and finally use all of the above as part of our toolkit for measuring efficiency and effectiveness improvements in Chapter 6.

1.2 Definitions

We introduce and discuss some terminology that may not be immediately familiar without considerable background. To support the reader, we provide brief definitions for how these terms are used in the context of this work.

Adversarial. In an adversarial setting, two (or more) systems interact, with one — the *attacker* — seeking to manipulate the input space for the *target* system, such that the input is valid but induces undesired behaviour. This manipulation can involve perturbing existing inputs or crafting entirely new inputs from scratch. The attacking system can be human, as in Chapter 3, machine, as in Chapter 5, or a collaboration between human and machine, as in Chapter 6.

Robustness refers to a system's ability to maintain consistent and accurate performance across a wide range of inputs, including those that are noisy or adversarial, or within the scope of but outside the distribution of the training data. A robust model is resilient to perturbations, ambiguities, or challenges in the input, such as typos, punctuation changes, unconventional phrasing, adversarial attacks, or domain shifts, while still producing reliable output without significant performance degradation. Three key aspects of robustness include *consistency* — the ability to produce stable and correct outputs across similar inputs, *reliability* — avoiding

catastrophic failures, and *out-of-distribution generalisation* — specifically referring to the long tail of phenomena that can be observed at training time within the scope of the task or set of tasks that the model was trained for.

Generalisation is a model's ability to maintain performance on new, unseen data by effectively applying learned functions beyond the training distribution. In the context of this work, we focus on evaluating generalisation specifically within the task of Reading Comprehension. However, as model capabilities have expanded to allow strong performance the multi-task setting, there is increased interest in generalisation encompassing cross-domain transfer — the ability to leverage capabilities gained in one domain or task and apply them to another.

The Loop. Here we refer to the full *iterative model development feedback loop*, encompassing: i) identifying the signal to train models on, typically captured as training data, ii) training a model, iii) evaluating or interacting with the trained model to form an understanding of its behavioural profile — both its strengths and weaknesses (including failure modes), and iv) iteratively refining by using the insights gained to improve both the training signal and algorithms for better performance. The focus on *humans-in-the-loop* presented in this work goes far beyond relying on human annotators for training data. In fact, Chapter 5 introduces a synthetic data approach which has become increasingly common in the era of post-training Large Language Models (LLMs). Rather, we highlight the importance of involving humans at all stages of the iterative model development process, ensuring consistent interaction with both data, models, and evals, providing iterative system performance improvements through human insight and feedback.

1.3 Thesis Overview & Structure

Following this introduction, the thesis is laid out as follows:

- In Chapter 2, we provide background and related work relevant to the material presented in this thesis, while that specific to each chapter is discussed in detail within the appropriate sections.
- In Chapter 3, we describe the published work "*Beat the AI: Investigating Ad-*

versarial Human Annotation for Reading Comprehension" which investigates the use of dynamic adversarial data collection for improving the robustness of extractive question answering models.

- In Chapter 4, we describe the published work "*Dynabench: Rethinking Benchmarking in NLP*" which introduces Dynabench, an open-source research platform for dynamic dataset creation and model benchmarking.

- In Chapter 5, we describe the published work "*Improving Question Answering Model Robustness with Synthetic Adversarial Data Generation*" which explores synthetic adversarial data generation for improving the robustness of question answering models and introduces adversarial human evaluation.

- In Chapter 6, we describe the published work "*Models in the Loop: Aiding Crowdworkers with Generative Annotation Assistants*" which investigates the use of generative assistants that interact with annotators to improve the efficiency and effectiveness of the data collection process.

- In Chapter 7, we conclude by highlighting our key findings and discussing future work.

1.3.1 Highlighted Contributions

Concretely, the contributions that this work makes are:

1. Provides in-depth analysis of the language understanding, comprehension and reasoning capabilities of contemporary RC models.

2. Identifies weaknesses and limitations of contemporary RC models.

3. Investigates methods for designing challenging datasets which push the current boundaries of RC model comprehension capabilities.

4. Investigates the collaborative and competitive aspects of involving humans and models in the annotation loop for improved system robustness.

5. Drives progress on general methodologies for more reliable, high signal, and granular system evaluation.

6. Develops and introduces methodologies for improving reading comprehension and reasoning capabilities of performant models.

1.4 Open-source Contributions

We also make various dataset, model and software platform contributions available publicly and open-source. These are discussed in detail within each chapter and we refer the reader to Chapter 7 for a summary of these contributions to the community.

1.5 Published Material

The material we will present in this thesis is based on a number of academic articles published at reputable conferences and journals. The linked narrative that will unfold over the next few chapters is based on four key contributions, but is influenced by other contributory published material. The material featured in Chapter 3 first appeared in:

- **Max Bartolo**, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating Adversarial Human Annotation for Reading Comprehension. In *Transactions of the Association for Computational Linguistics, 8:662–678.*

Many of the ideas around model robustness presented and discussed in this chapter have also been influenced or explored in:

- Johannes Welbl, Pasquale Minervini, **Max Bartolo**, Pontus Stenetorp, and Sebastian Riedel. Undersensitivity in Neural Reading Comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

- Maximilian Mozes, **Max Bartolo**, Pontus Stenetorp, Bennett Kleinberg, and Lewis D. Griffin. Contrasting Human- and Machine-Generated Word-Level Adversarial Examples for Text Classification. 2021. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

- Yao Lu, **Max Bartolo**, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. 2022. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

The work presented in Chapter 4 first appeared in or was otherwise influenced by:

- Douwe Kiela, **Max Bartolo**, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking Benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

- Tristan Thrush, Kushal Tirumala, Anmol Gupta, **Max Bartolo**, Pedro Rodriguez, Tariq Kane, William Gaviria Rojas, Peter Mattson, Adina Williams, Douwe Kiela. 2022. Dynatask: A Framework for Creating Dynamic AI Benchmark Tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Chapter 5 is based on the following publication:

- **Max Bartolo**, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. Improving Question Answering Model Robustness with Synthetic Adversarial Data Generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Finally, the material discussed in Chapter 6 was first presented in:

- **Max Bartolo**, Tristan Thrush, Sebastian Riedel, Pontus Stenetorp, Robin Jia, and Douwe Kiela. 2022. Models in the Loop: Aiding Crowdworkers with Generative Annotation Assistants. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

The topics of exploring the limits of annotator performance in both a training data and learning from human feedback setting, were further explored in:

- **Max Bartolo**, Hannah Kirk, Pedro Rodriguez, Katerina Margatina, Tristan Thrush, Robin Jia, Pontus Stenetorp, Adina Williams, and Douwe Kiela. Proceedings of the First Workshop on Dynamic Adversarial Data Collection. 2022. In *Association for Computational Linguistics, Seattle, WA, edition*.

- Tom Hosking, Phil Blunsom, and **Max Bartolo**. Human Feedback is not Gold Standard. 2024. In *The Twelfth International Conference on Learning Representations*.

Chapter 2

Background

> "That robot was created to run a Disinto on the moon. Its positronic brain was equipped for a lunar environment, and only a lunar environment. On Earth it's going to receive seventy-five umptillion sense impressions for which it was never prepared. There's no telling what its reactions will be. No telling!"
>
> — Robot AL-76 Goes Astray, Isaac Asimov

If the perturbation of a single component of a system's input, such as an environment variable, causes that system to malfunction in unexpected ways, then we have identified a mode of failure, knowledge of which will allow us to develop even more robust and generalisable systems. In this chapter, we introduce the discuss the task of Machine Reading Comprehension, including pivotal dataset contributions, modelling approaches, evaluation techniques and further historical context on the investigation of reasoning capabilities which motivate much of the work in this thesis. Following this, we briefly touch on ideas around algorithmic and ML system robustness in the context of interactions between humans and machines, then provide an overview of adversarial attacks and example creation strategies, and discuss some pertinent limitations. We finally provide relevant background on generative modelling approaches, improving model robustness, and human computer interaction — concepts that will regularly encounter throughout this thesis.

2.1 Machine Reading Comprehension

Formally, Machine Reading Comprehension is the task of providing an answer, a to a question, q about a context (or passage), c, modelled statistically as $p(a|q,c)$. In this general setting, the answer can take various forms such as a span extracted from the context, "Yes" or "No", a reference to a multiple-choice answer candidate, a free-form generation, and occasionally a response refusal. In order to achieve this reliably, a system is expected to require the capability to understand or comprehend the provided passage.

2.1.1 Overview and Historical Context

The ability of machines to comprehend language has been an area of research interest since the early days of modern computing (Turing, 1950). Among the early systems probing machine reading comprehension capabilities through question answering was the *Conversation Machine*, which was in part a response to Turing's work on machine thinking. While not formally set up as a Reading Comprehension task, this system was designed to respond to questions about the weather in conversation, requiring some elements of "reading" and "understanding" conversational context to do so. Even from these early days, concepts such as handling complex linguistic properties like negation, weighting relative relationships between question words, and forming semantic representations of both words and sequences were already being explored (Green et al., 1959).

In a review of fifteen early question-answering systems, Simmons (1965) offers insight into the diverse approaches that were already being implemented during the nascent stages of this research area. A notable example is *The Oracle*, which performed a syntactic analysis of the question and the text corpus containing the answer — one of the earliest recorded Machine Reading Comprehension tasks, where the example passages were simple sentences of the form "The teacher went to the school.", with the corresponding question "Where did the teacher go?". This system analysed parts-of-speech and aligned the question with the passage to extract the answer, in this case, "to school".

Other notable early work was *Protosynthex*, a system that captured the meaning

of words by comparing them with words of similar meaning based on how they appeared within encyclopedia articles. The idea of extracting word meanings based on how they co-occur in text drives today's most prominent semantic representation approaches — an example of an early synonym dictionary is shown in Table 2.1.

Word	*Words of Related Meaning*
animals	mammals, reptiles, fish
live	age
longer	older, ancient
men	person, people, women

Table 2.1: Example of the *Protosynthex* synonym dictionary for the question "What animals live longer than men?"

Other approaches introduced around this period that continue to broadly influence techniques in the development of modern question answering systems include query expansion, semantic correlation, and language-to-logical-form parsing (refer to Table 2.2) still commonly used in the development of Knowledge-Based Question Answering (KBQA) systems.

Question	*Logical Form*
When did humans first land on the moon?	$EventYear(FirstMoonLanding, ?x)$
What is the largest continent?	$argmax(\lambda x.continent(x), \lambda x.size(x))$

Table 2.2: Example of question transformation to logical forms for knowledge-based QA from Bobrow's *STUDENT* algebra problem solver (Bobrow, 1964).

Despite various limitations, these early question-answering systems laid the foundation for three major paradigms that power most modern question-answering systems. The first paradigm involves information retrieval-based methods, which identify salient passages of text within a corpus that are likely to contain an answer to the question. The second paradigm focuses on locating answers within a span of text — a task that requires comprehension capabilities and on which we focus for most of this work. The third paradigm concerns knowledge-base methods, where questions are transformed into logical forms to query relational or other struc-

tured databases and extract relevant answers. Early examples of modern question-answering systems, such as IBM Research's DeepQA architecture, employ a hybrid approach for improved performance. This approach powered the Watson system which also brought reading comprehension capabilities to the forefront of public perception by beating humans at the *Jeopardy* task (Ferrucci et al., 2010). While our work is generally motivated by driving improvements to real-world systems, we focus primarily on the Reading Comprehension paradigm as it requires a level of robust understanding where most previous systems often fail. However, many of the methodologies introduced in this work are general approaches that can be easily adapted to the other two paradigms.

Returning to the historical context, other notable contributions were the *LUNAR* system to capture linguistic constructions, term-level meanings, extensional inference and the application of semantic rules (Woods, 1978). During this era, a popular approach was to create sophisticated front ends for single-table databases, from which answers could be extracted. However, these database front-end systems often faced limitations, including domain constraints and portability issues. The *TEAM* system, introduced in 1987, was designed to address these portability issues.

Additionally, there was a challenge in handling language variations, as different users asked questions in diverse ways, and the system needed to be robust enough to understand these nuances. This led to a shift from a purely database-focused approach to a more dialogue-focused paradigm, as demonstrated by the *GUS* system developed by Bobrow for airlines in 1977. Dialogue-based systems aimed to capture the rationale and "real content" of questions by understanding both the user and the question itself. This shift was driven by the need to better handle real-world questions and improve the system's ability to capture the underlying intent and context (Jones, 2003).

Other significant contributions include *MASQUE* in 1993 (Androutsopoulos et al., 1993), *Webclopedia* in 2000 (Hovy et al., 2000), followed by *Mulder* in 2001 (Kwok et al., 2001) and *AnswerBus* in 2002 (Zheng, 2002) which was an open-domain QA system based on information retrieval.

More recent systems such as IBM's Watson represent a renewed focus on Reading Comprehension. In particular, Watson could only answer questions where the answer was contained within the passage. These ideas have also been expanded on in modern conversational agents such as Siri, Cortana, Alexa and Google Assistant which further seek to incorporate an element of dialogue understanding and extend question answering beyond a one-step interaction.

These system developments coincide with a shift towards a more data-centric definition of Reading Comprehension, providing a canonical task formulation and the creation of many datasets designed to help train ML models, which we will discuss in the next section.

Reading Comprehension has become somewhat synonymous with the task of Extractive Question Answering, and these terms have at times been used interchangeably (Gardner et al., 2019). We take the view that Reading Comprehension fundamentally requires the existence of text that a system is required to read and understand, in order to provide an answer to a question. Extractive Question Answering is a strictly more constrained definition of a task that also requires a system to demonstrate Reading Comprehension capabilities, but requires the answer to be a span extracted from the passage.

2.1.2 Datasets

2.1.2.1 TREC QA

The Text REtrieval Conference (TREC) question answering track, which started running in 1999, provided the first large-scale dataset requiring direct answers rather than a list of relevant documents that were likely to contain an answer (Voorhees and Tice, 1999). This provided a standard evaluation framework for the performance of different systems to be compared, as well as trained on.

The structural interpretation of a Question Answering task by TREC context differed considerably from earlier, predominantly database-driven approaches. One key difference was the source of information: TREC QA utilized general-purpose, open-domain text sources such as the *Wall Street Journal* and the *Foreign Broadcast Information Service*, although it was still restricted to the news and

web domains (Jones, 2003). This marked a shift from closed-domain, structured databases to more diverse and unstructured text sources. Questions in TREC QA were obtained from a combination of sources, including search engine logs such as *MSNSearch* and *AskJeeves*, as well as human-generated questions, reflecting a diverse range of query types and complexities.

As the TREC QA track evolved, slight structural changes were made, including the introduction of multiple correct answers for certain questions, until early 2005 (Voorhees et al., 2005). Another interesting attribute of TREC QA is the separation of factoid questions and list-based questions, where the expected response is a list of items, since a change in the data format in 2002. Typically, TREC QA provides a list of documents, a question (such as "What kind of animal is an agouti?"), the ID of the document containing the answer, and the answer string itself.

Due to variations in answers, correct evaluation involved matching responses with a given answer pattern (Voorhees and Tice, 2000). This ensured not only technical correctness but also identified the most appropriate answer in the context of the question. Consider the following question and answer patterns:

```
Q     What is a supernova?
A1    explod.*stars?
A2    calamitous death of a large star
A3    origin of gold
```

While technically correct on the basis of scientific theories pointing to heavy elements such as gold having originated within exploding supernovae (Cowan and Sneden, 2006), `A3` is not the expected answer given the question context, where the "most correct" answer is `A2`.

The TREC 8-13 tracks were further combined in 2007 to create a single comprehensive dataset (Wang et al., 2007). This consolidated dataset, QASent, quickly became a standard benchmark for question answering tasks, specifically for answer selection-based approaches.

2.1.2.2 MCTest

MCTest (Richardson et al., 2013) is a crowdsourced dataset designed to test machine comprehension of text, introduced and made freely available by Microsoft Research in 2013. It contains two subsets, differing by the methodology used to annotate original stories. *MC160* contains 160 manually curated stories, while *MC500* imposes further restrictions, requiring annotators to pass a grammar test and a reading comprehension test and doesn't rely on manual curation, containing an additional 500 stories.

MCTest is designed to evaluate machine performance on multiple-choice reading comprehension questions based on fictional stories. Intended to be open-domain, MCTest restricts its vocabulary and concepts to those understood by a 7-year-old child. The multiple-choice format also allows for simple and clear evaluation. The dataset comprises extracts from a corpus of fictional stories created by Mechanical Turk workers, covering a broad range of topics. Each story has four corresponding crowd-sourced questions (with an average of 7.77 words), each with four candidate answers, including the correct one.

To ensure that answering requires comprehension, annotators were asked to ensure that all answer options incorporated words from the stories and design the questions to require information from multiple sentences. This design aims to test the inference-making capabilities of reading comprehension systems, moving beyond simple keyword matching. While MCTest, offers a valuable resource for evaluation, its size is a limitation, with only 2,640 questions. This may pose challenges for training current ML models.

2.1.2.3 WikiQA

WikiQA, released in 2015 through a collaboration between the Georgia Institute of Technology and Microsoft Research, is a collection of 3,047 questions sampled use simple heuristics to identify question-like queries from Bing search engine logs between May 1, 2010, and July 31, 2011 (Yang et al., 2015). It provides answers candidates as sentences from the summary paragraphs of selected Wikipedia pages, along with crowdsourced annotations indicating whether the answer is contained

within the extract. For sentences provide an answer to the question, the dataset also provides an *answer phrase* annotation, defined as the shortest substring of the sentence that answers the question, similar to modern EQA datasets.

The WikiQA dataset contains 29,258 sentences, with 1,473 sentences labeled as providing answers to the corresponding questions. Statistical comparisons between the WikiQA and TREC QA datasets also reveal differences in question length, with TREC QA questions averaging 9.59 words and WikiQA questions averaging 7.18 words. Question length is likely influenced by the way questions are sourced (for example, crowdsourced versus search engine queries) and passage complexity (for example, the relatively low passage complexity of MCTest, with an average question length of 7.77 words).

There are also differences in the types of questions encountered, with the WikiQA questions representing the types of queries users naturally input into search engines, reflecting real-world query patterns and language usage. We highlight these differences in question type distribution in Table 2.3, in particular showing a considerably increased focus on descriptive questions in WikiQA.

Question Class	*TREC 8-13 QA*	*WikiQA*
Location	37 (16%)	373 (12%)
Human	65 (29%)	494 (16%)
Numeric	70 (31%)	658 (22%)
Abbreviation	2 (1%)	16 (1%)
Entity	37 (16%)	419 (14%)
Description	16 (7%)	1087 (36%)

Table 2.3: Comparison of the question class distributions across the TREC 8-13 QA (QASent) and WikiQA datasets.

Despite limitations in size and answer structure, WikiQA dataset made a valuable contribution to the area of machine reading comprehension, particularly in the context of answer span extraction.

2.1.2.4 CNN/Daily Mail

The CNN/Daily Mail reading comprehension datasets, introduced by Hermann et al. (2015), are based on news articles extracted from the *CNN* and *Daily Mail* websites, and designed to test machine ability to read real documents and answer complex questions. 313k articles were collected, dated until April 2015 and starting from April 2007 for *CNN*, and June 2010 for *Daily Mail*, along with corresponding article summaries and bullet point supplements. A corpus of over 1.3 million document-query-answer triples was created by converting the bullet point summaries into cloze-style questions, with an average of 12.5 and 14.3 words per question for the *CNN* and *Daily Mail* data respectively. Both these figures indicate longer questions than those previously discussed, likely an artefact of the data curation process.

In this process, detected entities were replaced with placeholders, and the system being tested was asked to fill in the blanks with the masked entity as the answer, demonstrating its reading comprehension abilities as shown in Figure 2.1. The LAMBADA dataset also shares structural similarities to CNN/Daily Mail, emphasizing the need for models to capture a broader context to effectively answer questions (Paperno et al., 2016). This cloze-style task format (Taylor, 1953) has various convenient properties that permit reading comprehension system evaluation using questions structured in an automated fashion and using minimal human supervision. These ideas have also been extended to build extractive QA systems trained on synthetic data obtained through cloze-style augmentation combined with question generation (Lewis et al., 2019).

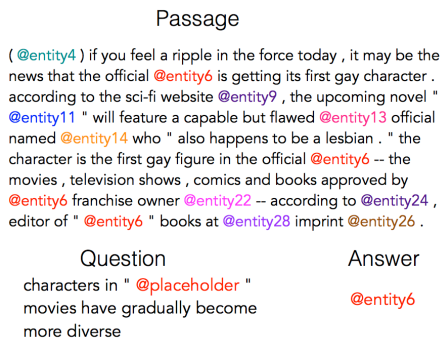


Figure 2.1: Example from the *CNN* dataset (Chen et al., 2016).

Various limitations have also been highlighted by Chen et al. (2016), including that the CNN/Daily Mail dataset is simpler than previously believed and that conventional NLP systems can perform better than expected in part due to an over-reliance on paraphrases due to the nature of the questions, that it primarily requires single-sentence relation extraction, falling short of larger-context text understanding, and that current systems perform well on unambiguous, single-sentence cases, leaving limited room for further improvement due to data preparation issues, such as coreference errors and entity anonymisation.

2.1.2.5 The bAbI Project

In late 2015, Facebook AI Research introduced a set of noiseless toy datasets as part of the bAbI project (Weston et al., 2015), aimed at evaluating reading comprehension systems' reasoning abilities through question answering tasks. The tasks cover a diverse range of scenarios, including identifying supporting facts, answering Yes/No questions, and handling queries related to reasoning over time, position, and size.

The *Children's Book Test* (CBT), a cloze-form dataset designed to examine the roles of context and memory in language understanding, was also introduced. This dataset is constructed using freely available books from Project Gutenberg, structured such that the first 20 sentences provide context, and the 21st sentence has a word removed, forming the query. Ten candidate answers are provided, with an average query length of 30.9 words — this is also an artefact of the way queries were constructed.

2.1.2.6 MS MARCO

The Microsoft AI & Research Machine Reading COmprehension (MARCO) dataset Nguyen et al. (2016) is a large-scale open-domain dataset of 100,000 question-answer pairs. It focuses on real-world data, with questions sourced from actual user queries and contexts scraped from documents retrieved using the Bing search engine. All answers are human-generated, and some questions have multiple valid responses.

Another aspect of note is the inclusion of noise, such as spelling or grammat-

ical errors and colloquialisms, reflecting real-world query patterns. Answers are not limited to specific spans from the text and may require merging different parts of the document to construct the correct answers, potentially requiring systems to demonstrate multi-hop reasoning capabilities for good performance. This approach also addresses the question bias inherent in crowd-sourced datasets and presents a more realistic and challenging scenario for evaluating question-answering systems, although it carries the limitations previously discussed around using questions sourced from search engine queries.

2.1.2.7 SQuAD1.1

The Stanford QUestion Answering Dataset (SQuAD) version 1.1 is a comprehensive collection of 107,785 crowdsourced question-answer pairs (Rajpurkar et al., 2016) derived from 536 high-quality Wikipedia articles, sampled uniformly at random from the top 10,000 articles sorted by Wikipedia's internal PageRanks metric. Answers are provided as text spans extracted from the source articles, with an example shown in Figure 2.2.

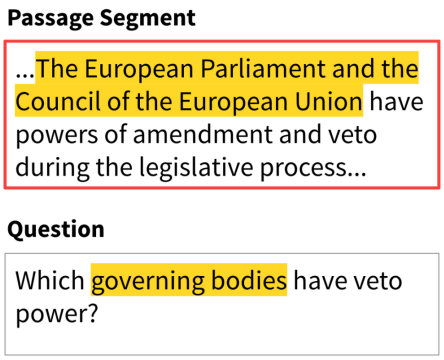


Figure 2.2: Example from SQuAD1.1 showing the answer as a span from the passage and requiring external knowledge (that the European Parliament and the Council of the European Union are governing bodies) in order to answer the question.

Additional answers were collected per question for validation, as well as to estimate human performance. Human performance, estimated by taking the maximum across each pair of candidate answers on the test set, is given at an F1 score of 91.2% and exact match score of 82.3%. These evaluation methods are discussed in detail in the subsequent sections, along with a more thorough analysis of the types

of reasoning required to answer questions in SQuAD1.1 and which are also further investigated in Chapter 3.

2.1.2.8 SQuAD2.0

SQuAD2.0 (Rajpurkar et al., 2018) is among the first research efforts to directly target poor Reading Comprehension model robustness, which we shall also discuss at a later stage. Extractive Question Answering systems can often locate the correct answer to a question in a passage, but they also tend to make unreliable guesses on questions for which the correct answer is not stated in the context, or get easily distracted by irrelevant information (Jia and Liang, 2017).

SQuAD2.0 is an updated version of the Stanford Question Answering Dataset, combining the previous SQuAD1.1 dataset with over 50,000 unanswerable questions. These unanswerable questions are adversarially crafted by crowdworkers to resemble answerable questions, challenging models to determine when no answer is supported by the provided paragraph and abstain from answering. This is designed to help address limitations of existing datasets, which either focus solely on answerable questions or use automatically generated unanswerable questions that are easily identifiable.

This resource serves as a more rigorous test of natural language understanding, as demonstrated by the performance gap between SQuAD1.1 and SQuAD2.0 for strong systems at the time with a best F1 score of 86% on SQuAD1.1 compared to 66% on SQuAD2.0. It also illustrates a significant performance gap between human accuracy and existing models, similar to SQuAD1.1 at its time of release. This updated version of the dataset requires them to exhibit a deeper understanding of the context. It further encourages the development of more robust and discerning question-answering systems, capable of making informed decisions about when to provide an answer and when to refrain from answering. The work presented in Chapter 3 can be seen from the perspective of providing a further updated version of this line of research.

Finally, it is worth making a quick note of distributional mismatch of negatives between the training set and validation and test sets in SQuAD2.0, as this is

often overlooked. The training set has train data has roughly twice as many answerable questions as unanswerable ones, however, the validation and test sets exhiit a roughly one-to-one ratio of answerable to unanswerable questions. To some extent then, this dataset also tests the ability of systems to generalise across train/test distributional mismatch.

2.1.2.9 NaturalQuestions

Natural Questions (NQ) is a question answering dataset from Google Research (Kwiatkowski et al., 2019). It contains real user questions issued to the Google search engine, and answers found from Wikipedia by annotators. Questions consist of real anonymised, aggregated queries issued to the Google search engine, and paired with Wikipedia pages from the top 5 search results.

Answers are annotated as a long answer (typically a paragraph) and a short answer (one or more entities) if present on the page, or null if no answer is present (similar in motivation to SQuAD2.0. NQ uses naturally occurring queries and focus on finding answers by reading an entire page, rather than extracting answers from a short paragraph, and addresses criticism that SQuAD and similar datasets received around lack of diversity due to annotator-written questions being conditioned by the passages they were provided to read — although we will find in Chapter 3 that doesn't go particularly far from the perspective of increasing question diversity.

2.1.2.10 DROP

A Reading Comprehension dataset requiring Discrete Reasoning Over Paragraphs (DROP) was introduced in 2019 by researchers at the University of California, Irvine, Peking University, and the Allen Institute for Artificial Intelligence (Dua et al., 2019). It also seeks to address some of the brittleness issues of existing systems, focusing in particular on numerical and discrete reasoning such as addition, counting, or sorting. With nearly 100k examples, it also provided a valuable training resource. DROP also introduced structured answer types such as dates or numbers, as well as span extracts, and included an adversarial component that encouraged annotators to ask questions that the model-in-the-loop, BiDAF (Seo et al., 2017), struggled to provide a valid answer to.

A limitation here was that BiDAF is an extractive model and DROP's answer type flexibility meant that for many of the questions asked, the model in the loop could not predict the correct answer as the answer was not a span within the passage. This also provided annotators with a relatively simple strategy for creating adversarial examples by asking questions requiring particular answer types. We build on this extensively in Chapter 3.

2.1.3 Modelling Approaches

One of the early high-performing RC models was the Bi-Directional Attention Flow (BiDAF) model (Seo et al., 2017), which we have previously mentioned earlier as the model-in-the-loop for DROP, and which we use as the baseline model-in-the-loop in Chapter 3. In this context, we provide an overview of its architecture.

BiDAF is a hierarchical multi-stage architecture for machine comprehension. It consists of six layers: character embedding, word embedding, contextual embedding, attention flow, modelling layer, and output. The character and word embedding layers represent each word as a fixed-size vector, using CNNs and pre-trained GloVe vectors, respectively. These embeddings are fed into an LSTM contextual embedding layer to capture temporal and sequential information.

The attention flow layer computes bi-directional attention, allowing it to flow to the next layer, and capturing the relevance of query words to context words and vice versa. The modelling layer uses a two-layer bi-directional LSTM to capture interactions between context words conditioned on the query. Finally, the output layer predicts the start and end indices of the answer phrase in the passage by calculating the independent probability distributions of each over the full length of the context. The probability of each word being the start, s, of the answer is:

$$\mathbf{p}^s = \text{softmax}(\mathbf{w}_{\mathbf{p}^s}^\top [\mathbf{G}; \mathbf{M}])$$

Where $\mathbf{w}$ represents the weights of the output layer, $\mathbf{G}$ is the matrix of combined contextual embeddings and attention vectors from the previous layers and $\mathbf{M}$ is the matrix representing the state of the modelling layer.

The probability that each word corresponds to the end, e, of the answer phrase is calculated in a similar manner, and there is a final aggregation step that sorts all possible candidate answers by their joint answer probability calculated as the product of their start and end probabilities, excluding invalid ones, that is those for which the end index is earlier in the text than the start index. The highest ranked index pair (s, e) is then used to decode the answer text.

Models based on a Masked Language Modelling (MLM) pretraining objective using the transformer architecture, such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b) and ELECTRA (Clark et al., 2020), have also been adapted for Extractive Question Answering and demonstrated substantial performance improvements over earlier systems.

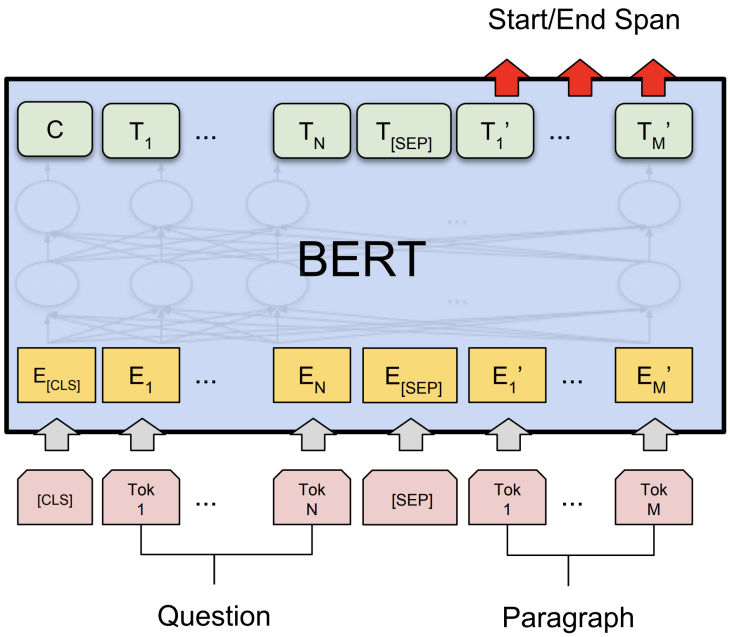


Figure 2.3: Illustration of the adaptation of BERT for Extractive Question Answering reproduced from Devlin et al. (2019) showing the start and end canddiate predictions corresponding to input tokens in the paragraph.

As shown in Figure 2.3, two output layers, one corresponding to the start and one corresponding to the end of the predicted answer span, take the token representations corresponding to the passage as inputs. Similar to the independent calculation under validity constraints described for BiDAF, the probability of word at index i being the start or end of the answer span is computed as a dot product between the

token representation at that position and the corresponding output vector, followed by a *softmax* over all of the words in the passage.

While it has been suggested that information about which index has been selected as the start of the answer should influence the selection of which index should end the answer phrase, this has not been extensively studied. More recently, generative (also referred to as text-to-text) models that predict free text as output have also been shown to give good performance on Extractive Question Answering tasks (Raffel et al., 2020).

2.1.4 Evaluation

Two commonly used Extractive Question Answering evaluation measures, popularised in part by the ease of access of the SQuAD1.1 implementation, are Exact Match and word-overlap F1 Score calculated on pre-processed ground truth answers and model predictions. We introduce and define them here since they are used throughout this work to evaluate system performance.

2.1.4.1 Exact Match Score

Exact Match indicates if a predicted answer exactly matches any of the candidate ground truth answers after normalisation and post-procesing. Normalisation typically includes lowercasing, converting non-unicode characters to their unicode counterparts, removing articles such as “a”, “an”, and “the”, stripping punctuation and adjusting for white-space disparities.

The system’s overall Exact Match score is calculated by taking the mean across sample match indicators (a per-sample score of 0 or 1), typically presented after being converted to a percentage.

2.1.4.2 F1 Score

F1 score is a measure of word overlap between a model-predicted answer and the ground truth after normalisation. For datasets which provide multiple valid ground truth answers, such as SQuAD, the reported score is the the maximum F1 calculated over all valid ground truth answers. Comparing systems across datasets with varying numbers of ground truth answers could provide potentially misleading results,

particularly in the presence of short passages which are more likely to a have a relatively small set of possible answer candidates. To fairly draw direct comparisons, where appropriate we evaluate model predictions against the majority voted ground truth answer as described in Chapter 3.

F1 score is formally defined as the harmonic mean of precision and recall, where precision, P, is the number of overlapping words between the predicted and ground truth answer phrases divided by the number of words in the predicted phrase, and recall, R, is the word overlap divided by the length of the ground truth phrase. F1 score is then defined as:

$$\text{F1} = \frac{2PR}{P+R}$$

Similar to the computation for exact match, the reported system performance score is the macro-average, or average across samples, of sample-level F1 scores.

While F1 score is a generally better indicator of answer correctness than exact match, it does not perfectly capture the correctness of a predicted answer since it is sensitive to instances where an answer phrase could be missing or have additional words that have little significance, such as articles or prepositions. This remains a challenge with automated evaluation approaches, and was part of the reason we migrated from using F1 score to judge whether a human annotator had successfully beaten the AI in Chapter 3 to human-judged and validated measures of success in the subsequent chapters.

2.1.5 Reasoning Capabilities

Understanding written text is a formidable task and requires a complex combination of skills including determining explicit sentence meaning, making inferences about their likely implicit meaning, and inferring the implicit connections between sentences (Norvig, 1986). In the context of language understanding, and specifically machine reading comprehension, Bottou (2014) suggests a plausible definition for machine *reasoning* as "algebraic manipulation of previously acquired knowledge in order to answer a new question".

The reading comprehension task formulation provides a convenient test-bed for machine reasoning capabilities both due to the unrestricted diversity of possible questions and complexity of source passages, and this was recognised from the early days of large-scale RC dataset construction.

QA4MRE (Peñas et al., 2011; Sutcliffe et al., 2013), an early multiple-choice RC evaluation dataset track, was designed to contain questions covering five different question types: purpose, method, causal, factoid, and which-is-true — with factoid questions further divided into: Location, Number, Person, List, Time and Unknown. See Table 2.4 for examples.

Question Type	**Example**
PURPOSE	What is the aim of protecting protein deposits in the brain?
METHOD	How can the impact of Arctic drillings be reduced?
CAUSAL	Name one reason why electronic dance music owes a debt to Kraftwerk.
FACTOID (number)	What is the approximate number of TB patients?
WHICH-IS-TRUE	Which problem is similar in nature to global warming?

Table 2.4: Examples of questions of different types from QA4MRE.

Questions further required a variety of inference capability including: linguistic inferences such as co-reference and deictic references (like "then" and "here"), ontological inferences (such as part-of relations), inferences on causal relationships or procedural steps, and composite inference requiring an answer to be formulated by considering different parts of the passage and optionally background knowledge. Background knowledge is further categorised into four types: specific document-related facts (e.g. the relationship between two people mentioned in the document), general facts (such as geographical knowledge, acronyms and unit conversion), general abstractions used for interpreting language or implicit knowledge (e.g. knowing that petroleum companies drill wells), and linguistic knowledge (such as synonyms or word-to-number conversions). Around a third of questions in the test set also had no valid answer.

Test set questions were further segregated into simple, intermediate, and difficult questions, aiming for a balanced distribution of question difficulty across test documents. These were defined as:

- *Simple*: the answer and the fact questioned could be found in the same sentence in the document.

- *Intermediate*: the answer and the fact questioned were not in the same sentence and could be several sentences apart (multi-sentence reasoning).

- *Difficult*: require utilising information from the background collection.

Steps were also taken to reformulate or paraphrase questions to minimise word overlap between the document and the question, a potentially strong signal for locating plausible answer candidates. The inference type required to answer each question was tracked to facilitate analyses regarding which types of questions were difficult for the systems, and why.

As a result of the difficulty in judging whether a correct answer had been chosen at random or derived from a valid process of deduction, a further evaluation experiment was carried out in the third year of the task. This involved the creation of Auxiliary questions which were derived from an original question, but with an added simplification deliberately removing one inference step — achieved through hypernym replacement, noun phrase synonymy, or verbal entailment. The being that if a system answered a question incorrectly but the corresponding Auxiliary question correctly, it suggests that the system was near to answering the question but could not perform the inference step.

The expert-annotation approach which permitted such in-depth analysis and consideration of reasoning phenomena, also limited the QA4MRE datasets in terms of size. This was a key challenge that MCTest (Richardson et al., 2013) aimed to overcome through the application of crowd-sourcing. It consisted of a challenge dataset of 2000 questions on 500 fictional stories, an order of magnitude larger than QA4MRE, yet still considerably small by current standards.

Despite the very different annotation approach, MCTest also gave consideration to testing advanced RC abilities such as causal reasoning and understanding world knowledge. While the design focused on short passages rather than matching against existing knowledge bases, at least 50% of the questions required processing

a minimum of two sentences from the passage to answer, designed to encourage multi-sentence reasoning.

Synthetic dataset construction catering to particular reasoning skills is another approach to requiring reasoning and natural language understanding in a question answering setting. The QA subset of the bAbI tasks aims to categorize different kinds of questions into skill sets, which are then used to define the tasks. These question skill sets include: factoid questions with one, two or three supporting facts, two and three argument relations requiring the differentiation of entities in the text, Yes/No questions, counting, reasoning with lists or sets, simple negation, basic and compound coreference, conjunction, basic deduction and induction via inheritance or potential inheritance of properties, reasoning about time, position and size, path finding, and reasoning about motivation (Weston et al., 2015).

Observations that large amounts of real data and simpler models tended to outperform more elaborate models based on less data (Halevy et al., 2009), together with the linguistic limitations of the synthetic approach, drove the construction of large-scale datasets with human-generated questions such as SQuAD (§2.1.2.7). This large scale inhibits reasoning type annotation on a per-question basis as in QA4MRE, however, the authors do provide a qualitative reasoning type analysis by manually labelling a sample of 192 questions. They find that 64.1% of questions involve syntactic variation in which the dependency structure of the question is different to the answer sentence in the passage, 33.3% involve lexical variation (synonymy) or paraphrasing, 13.6% require multiple sentence reasoning, 9.1% involve lexical variation requiring world knowledge, and 6.1% are ambiguous.

Subsequent large-scale RC datasets commonly perform a similar qualitative analysis of the question reasoning requirements, typically around a custom taxonomy of reasoning types adapted to the specific dataset design considerations.

NarrativeQA (Kočiský et al., 2018), a dataset of 46,765 human generated questions based on summaries of books and movie scripts, finds that only a small number of questions and answers are shallow paraphrases and that most questions require reading segments at least several paragraphs long, and in some cases even multiple

segments spread throughout the story.

Work following the introduction of the CNN/Daily Mail dataset (Hermann et al., 2015) constructs a tailored taxonomy for the task and provides a detailed analysis (Chen et al., 2016). The authors define six reasoning categories as follows:

Exact match: The nearest words around the placeholder are also found in the passage surrounding an entity marker; the answer is self-evident.

Sentence-level paraphrasing: The question is entailed or rephrased by exactly one sentence in the passage, such that the answer can be exactly and only identified from that particular sentence.

Partial clue: Despite no complete semantic match between the question and some sentence, we are still able to infer the answer through partial clues, such as some word/concept overlap.

Multiple sentences: Requires systems to process and understand multiple sentences to infer the correct answer.

Coreference errors: Includes examples with critical coreference errors for the answer entity or key entities appearing in the question.

Ambiguous or very hard: Includes examples for which the authors think that humans are not able to confidently obtain the correct answer.

This work includes a qualitative analysis of the CNN dataset and finds that 41% require paraphrasing, 19% provide a partial clue, 17% are ambiguous/hard, 13% are an exact match, 8% involve coreference errors, and 2% require reasoning over multiple sentences. The work also provides a per-category performance of a neural RC system based on the Attentive Reader model (Hermann et al., 2015) and find that the model performs well on exact match (100%), paraphrasing (95.1%) and partial clue (89.5%), but struggles on questions requiring multiple sentence reasoning (50%), coreference (37.5%) or questions in the ambiguous/hard category (5.9%). While insightful, this analysis is carried out on a very small sample size, with there only being 2 examples of questions requiring multi-sentence reasoning, for example.

In NewsQA (Trischler et al., 2017), a dataset of over 100,000 human-generated

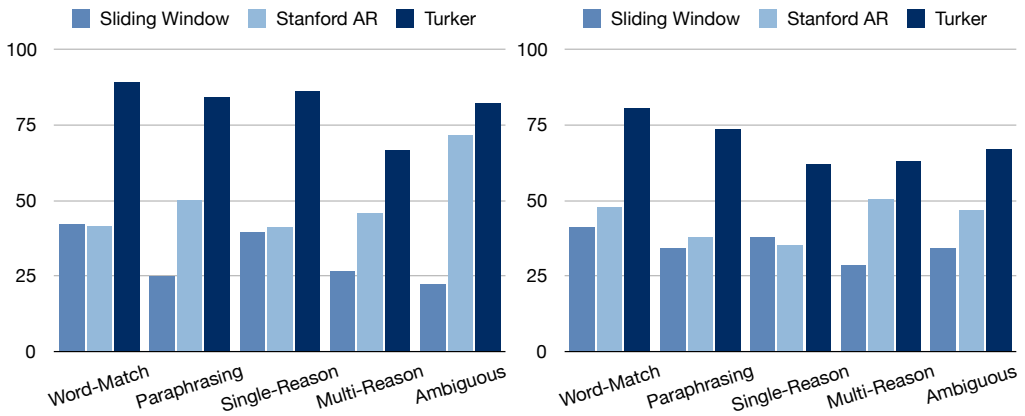
question answer pairs on a set of over 10,000 CNN news articles, the authors stratify reasoning types based on a variation of the taxonomy presented by Chen et al. (2016), essentially joining *partial clue* and *coreference errors* into a single category *inference*, and referring to *multiple sentences* as *synthesis*. They perform a thorough qualitative analysis of 1,000 questions sampled from the validation sets of both NewsQA and SQuAD1.1. Their findings are shown in Figure 2.4.

Reasoning	Example	Proportion (%) *NewsQA*	*SQuAD*
Word Matching	Q: **When were** the **findings published**? S: Both sets of research **findings were published Thursday**...	32.7	39.8
Paraphrasing	Q: **Who** is the **struggle between** in Rwanda? S: The **struggle pits ethnic Tutsis**, supported by Rwanda, **against ethnic Hutu**, backed by Congo.	27.0	34.3
Inference	Q: **Who** drew **inspiration** from **presidents**? S: **Rudy Ruiz** says the lives of US **presidents** can make them **positive role models** for students.	13.2	8.6
Synthesis	Q: **Where** is **Brittanee Drexel** from? S: The mother of a 17-year-old **Rochester**, **New York** high school student ... says she did not give her daughter permission to go on the trip. **Brittanee** Marie **Drexel**'s mom says...	20.7	11.9
Ambiguous/Insufficient	Q: **Whose mother** is **moving** to the White House? S: ... **Barack Obama's mother-in-law**, Marian Robinson, will **join** the Obamas at the **family's private quarters** at 1600 Pennsylvania Avenue. [Michelle is never mentioned]	6.4	5.4

Figure 2.4: Reasoning mechanisms needed to answer questions in NewsQA and SQuAD1.1 (Trischler et al., 2017).

RACE (Lai et al., 2017), a benchmark evaluation dataset from English exams for middle and high school Chinese students between 12 to 18 years of age and consisting of approximately 100,000 multiple-choice questions generated by English instructors, also builds on this reasoning taxonomy. RACE is further divided into RACE-M collected from examinations designed for 12-15 year-old middle school students, and RACE-H from examinations designed for 15-18 year-old high school students. Reasoning type analysis on 1,000 samples of each show a substantial reduction in word matching between RACE-M (29.4%) and RACE-H (11.3%) — both lower than SQuAD1.1 and NewsQA, and similar to CNN/Daily Mail (13%) for RACE-H. There is a notable increase in proportions for the paraphrasing, single-sentence reasoning, multi-sentence reasoning and ambiguous types between RACE-M and RACE-H, particularly with regards to the latter with 1.8% for RACE-M and 7.1% for RACE-H of questions deemed ambiguous. The results

are shown in Figure 2.5.



(a) RACE-M (b) RACE-H

Figure 2.5: Test accuracy by reasoning type on RACE (Lai et al., 2017).

The work further provides performance analysis for two different reading models: i) based on a sliding window algorithm of TFIDF scores between the concatenated question and answer, and windows from the passage, and ii) the Stanford Attentive Reader (Chen et al., 2016). This analysis is compared to that of non-expert humans, revealing a degradation in performance as reasoning requirements increase.

TriviaQA (Joshi et al., 2017), a dataset of over 650K distantly-supervised question-answer-evidence triples, presents a reasoning taxonomy similar to that of SQuAD1.1, with the addition of *Lists/Table*. In the case of Wiki documents, the authors find that 69% of questions involve syntactic variation, 41% of questions involve lexical variation in terms of synonyms, 40% require reasoning over multiple sentences — more than three times as much as in SQuAD1.1, 17% require common sense or external knowledge — nearly double the amount of SQuAD1.1, and 7% have answers found in tables or lists. Examples in TriviaQA exhibit more lexical and syntactic variance than in SQuAD1.1, indicating that decoupling the question generation process from source passage selection may results in more challenging questions. NaturalQuestions (Kwiatkowski et al., 2019) similarly decouples question generation from passage selection by sourcing questions from search queries

(as in WikiQA (Yang et al., 2015) and MS MARCO (Nguyen et al., 2016)), however, no qualitative reasoning type analysis is provided which would allow further investigation of this observation.

As previously discussed, motivated in part by findings that models can do remarkably well on SQuAD1.1 by exploiting context and type-matching heuristics, both on SQuAD1.1 and NewsQA (Weissenborn et al., 2017), and that success on SQuAD1.1 does not ensure robustness to adversarially-created distracting sentences (Jia and Liang, 2017), SQuAD2.0 requires models to determine whether or not a question is answerable from the passage. In this work, the authors also provide a breakdown of reasoning types for the negative or unanswerable examples. They find that 24% are cases where the paragraph does not contain the information required to answer the question, 21% involve an entity swap or replacement, 20% involve antonym use, 15% involve a word or phrase which mutually excludes a condition to which an answer does exist, 9% require an understanding of negation, 7% are in fact answerable (i.e. noise in the data), and 4% require a condition to be met that is not satisfied by the paragraph (i.e. impossible condition).

The AI2 Reasoning Challenge (ARC) dataset (Clark et al., 2018) consists of natural, grade-school science multiple-choice questions and is designed to require more powerful knowledge and reasoning than previous challenges such as SQuAD. The ARC Challenge set, in particular, contains questions adversarially filtered through an Information Retrieval (IR) solver and a Pointwise Mutual Information (PMI) solver. The authors define a taxonomy of knowledge types tailored to science questions and based on a sample of 100 questions from the Challenge Set, find that 17.9% are basic facts, 17.9% are definitions, 16.1% require knowledge of processes, 14.3% require knowledge of experiments, 10.7% are about structure, 10.7% require knowledge of function or purpose, 7.1% require algebraic reasoning, and 5.4% require spatial or kinematic understanding.

The authors of DROP (Dua et al., 2019), also discussed earlier, construct a reasoning taxonomy tailored to the dataset, and find that subtraction is a required skill for 28.8% of questions, followed by selection (19.4%), comparison (18.2%),

addition (11.7%), sorting (11.7%), counting (16.5%), that 6% of answers are a set of spans, co-reference resolution (3.7%), other arithmetic reasoning (3.2%), and that some other form of reasoning is required for 6.8% of questions. The authors further perform an error analysis on 100 incorrect predictions from the NAQANet model and find that the most common errors occur on questions requiring complex reasoning such as arithmetic operations (in 51% of errors), counting (30%), domain knowledge and common sense (23%), co-reference (6%), or a combination of reasoning types (40%).

CODAH (Chen et al., 2019), an adversarially constructed evaluation dataset of commonsense reasoning, provides a question categorisation taxonomy including idioms, negations, polysemy, reference and questions requiring quantitative reasoning. However, a distribution of question categories finds that 75.3% of questions do not belong do either of these categories, possibly indicating an increased diversity of questions collected through the adversarial construction approach. A class-wise accuracy analysis finds that humans perform consistently well (¿ 90%) on all question types, with BERT (Devlin et al., 2019) and GPT-1 (Radford et al., 2019) performing worst on quantitative (58.2%, 48.7%), followed by polysemy (63.9%, 56.2%) and negation (65.2%, 62.4%), and perform best on idioms (71.4%, 71.9%) and reference (72.4%, 70.7%).

In the context of commonsense reasoning, CosmosQA (Huang et al., 2019) provides a taxonomy of commonsense reasoning types including pre-/post-conditions — causes/effects of an event, motivations — intents or purposes, reactions — possible reactions of people or objects to an event, temporal events — what events might happen before or after the current event, situational facts — facts that can be inferred from the description of a particular situation, counterfactuals — what might happen given a counterfactual condition, and other — other types, e.g., cultural norms. ReCoRD (Zhang et al., 2018) provides a taxonomy similar to those previously observed, and with 75% of questions requiring commonsense reasoning, breaks these down into conceptual knowledge, causal reasoning, naive psychology (involving predictable human mental states in reaction to events), and other (such

as social norms, planning, spatial reasoning).

CoQA (Reddy et al., 2019), a conversational question answering challenge dataset, provides a breakdown of linguistic phenomena for the relationship between a question and its conversation history centred around coreference, and also introduces the phenomenon of *pragmatics* — similar in definition to the previously discussed inferential or implicit reasoning — finding that 27.2% of questions presented this phenomenon. The authors further find that there was a lexical match between the question and passage in 29.8% of dataset instances and paraphrasing in 43.0% of cases.

Also in the context of conversational question answering, Ramos and Lipani (2024) extend the ShARC dataset (Saeidi et al., 2018) with automatically extracted explanations designed to capture the type of reasoning behind the model's output for over 24k conversations, the utility of which is further validated by experts. Such approaches could provide a standard methodology for the easy comparison of reasoning types across datasets, and help identify new patterns in model behaviour that could further be used to drive modelling improvements.

2.2 Robustness in an ML Context

This thesis focuses on robustness in the context of ML systems, but the term is borrowed from traditional software engineering and algorithmic design. In the ML context, building on work on robust optimisation, robustness has been defined as *the property that if a testing sample is "similar" to a training sample, then the testing error is close to the training error* (Ben-Tal and Nemirovski, 1998, 1999; Bertsimas and Sim, 2004) — a definition that has also been used to define the generalisation of ML systems based on their robustness (Globerson and Roweis, 2006; Xu and Mannor, 2010).

ML system robustness generally requires a lack of *sensitivity* to disturbances in the input space — irrespective of whether these occur to numeric, or text representations of the input for models processing language. Robustness further spans both over- and under-sensitivity, meaning that system outputs should both not fluc-

tuate massively when minor changes do not require them to, such as when adding a typo to an input, but should also change considerably when minor input perturbations do require them to, for example by introducing breaking semantic changes such as by changing the entity about which a question is asked (Welbl et al., 2020). Recent work also seeks to define evaluation methodologies for ML system robustness (Carlini and Wagner, 2017b), and understand the trade-offs between improving system robustness and accuracy under adversarial training (Tsipras et al., 2019), although whether and how these system objectives conflict remains disputed (Stutz et al., 2019). What is certain is that both are important goals for the development of useful real-world systems.

2.3 Adversarial Examples

We have briefly touched on the concept of adversarial examples in earlier discussions. The concept of adversarial examples in ML originated in the computer vision space, where small, carefully crafted perturbations to input data could cause misclassification by machine learning models (Globerson and Roweis, 2006). Goodfellow et al. (2015) showed that imperceptible noise, when added to images, could lead deep neural networks to misclassify with high confidence. This finding sparked a flurry of interest in understanding and defending against these "adversarial attacks".

Initial efforts focused primarily on developing defensive mechanisms, such as adversarial training, where models are exposed to adversarial examples during training to improve their robustness (Goodfellow et al., 2015; Szegedy et al., 2015; Kurakin et al., 2017b; Raghunathan et al., 2018; Yuan et al., 2019). For a thorough review of adversarial attacks and defence mechanisms, see Wiyatno et al. (2019).

Developing robust models was soon realised to be more challenging than anticipated. In 2017, Athalye et al. (2018a) demonstrated that many proposed defences were ineffective against stronger, more sophisticated attacks. Adversarial examples were also found to be challenging to detect, easily bypassing detection systems (Carlini and Wagner, 2017a). This highlighted the need for a deeper understanding of the underlying model vulnerabilities. As the field matured, so too did

the sophistication of adversarial attacks, along with a focus on real-world adversarial examples (Kurakin et al., 2017a; Hendrycks et al., 2021b).

More complex and subtle ways to manipulate inputs, moving beyond simple noise addition were explored such as leveraging gradient information (Carlini and Wagner, 2017b) or the transferability properties of adversarial examples – that is the their tendency to remain effective across different models – in a black-box setting without access to the model's parameters (Papernot et al., 2016).

With the increasing popularity and success of deep learning in NLP, the concept of adversarial examples was further extended and has been studied extensively in NLP as a technique for probing the limitations and identifying failure points of NLP systems and highlight model vulnerabilities — see Zhang et al. (2019) for a recent survey. The unique characteristics of text data, however, presented new challenges as, while image perturbations are often measured by their imperceptibility, textual adversarial examples also needed to consider linguistic fluency and semantic consistency. One of the earliest works in this direction was by Jia and Liang (2017), who proposed an attack that inserted distractor sentences to SQuAD for evaluating Reading Comprehension systems. This exploration also extended to other NLP tasks such as text classification (Ebrahimi et al., 2018b) and machine translation (Belinkov and Bisk, 2018).

A particular challenge in the NLP space is posed by the discrete search space, where altering a single word can change the semantics of an instance or render it incoherent. Recent work overcomes this issue by focusing on simple semantic-invariant transformations, showing that neural models can be *oversensitive* to such input perturbations. For instance, Ribeiro et al. (2018) use a set of simple perturbations such as replacing "Who is" with "Who's". Other semantic-preserving perturbations include typos (Hosseini et al., 2017), the addition of distracting sentences (Jia and Liang, 2017; Wang and Bansal, 2018), character-level adversarial perturbations (Ebrahimi et al., 2018b), and paraphrasing (Iyyer et al., 2018). Wallace et al. (2019a) further identify universal attacks that are transferable across both examples and models.

There was also considerable work exploring methods for generating adversarial examples (Xiao et al., 2018; Baluja and Fischer, 2018; Alzantot et al., 2018; Athalye et al., 2018b) within and beyond the NLP domain. In Chapter 5, we extend these concepts by introducing a methodology to synthetically generate adversarial examples designed to emulate the attack strategies taken by human adversaries.

Generation of adversarial examples against generative models has also been explored (Kos et al., 2018), as well as work exploring the interactions and dependencies between adversarial examples and model robustness (Xie et al., 2020; Cisse et al., 2017; Ilyas et al., 2019).

2.4 Generative Language Models

We make use of generative models primarily in Chapter 5 to generate synthetic questions designed to be adversarial to performant models in-the-loop, and in Chapter 6 to assist human annotators with creating more effective adversarial attacks.

Generative models are a type of ML model that can generate new data by sampling from the distribution learnt on the underlying distribution of the training data. In NLP, they have been used for tasks such as language generation, text completion, and dialogue systems. One of the earliest and most widely used generative models in NLP is the n-gram language model, which predicts the probability of a word given its preceding words in a sentence. For a comprehensive historical overview, we refer the reader to Jurafsky and Martin (2000).

More recently, deep learning models, such as Recurrent Neural Networks (RNNs) (LeCun et al., 2015) and transformer-based models (Vaswani et al., 2017), have been used for generative tasks in NLP. Notable examples include sequence-to-sequence (seq2seq) models (Sutskever et al., 2014), which are commonly used for machine translation and text summarisation tasks, and the GPT family of models, which use transformer-based architectures to generate human-like text (Radford et al., 2018; Brown et al., 2020).

In this work, we use generative seq2seq models built on the transformer architecture. BART (Lewis et al., 2020), is pretrained as a denoising autoencoder,

trained by corrupting original text and learning to reconstruct it — it can be seen as generalising BERT as the encoder and the GPT decoder architecture. We finetune the pretrained BART model for the task of adversarial question generation.

We also make a quick note about the evaluation challenges posed by generative language models. Since the output space is relatively unconstrained, limited only by the model's vocabulary, such models can predict any output. As such, automated evaluation techniques, such as ROUGE for summarisation or BLEU for machine translation, tend to struggle to capture a reliable evaluation of model output.

2.5 Human-Computer Interaction

While our work involves various aspects of involving humans and models in the annotation loop, we touch briefly on ideas centres around Human-Computer Interaction (HCI), providing insight into how human annotators and generative assistants interact in Chapter 6.

HCI is an interdisciplinary field that focuses on design-specific user-interface technologies, specifically the interfaces where people and computers interact. It involves the study of how people interact with computer systems and the design of interfaces that are intuitive, efficient, and user-friendly. A primary goal of HCI research is to improve the usability and accessibility of computer systems by understanding the needs and capabilities of human users. This field draws on principles and methods from various disciplines, including computer science, cognitive psychology, human factors engineering, and user experience design, to create interactive technologies that are effective and easy to use (Preece et al., 1994; Myers et al., 1996; Carroll, 1997; Te'eni et al., 2005).

For a brief historical perspective, we refer the reader to Myers (1998) and for an in-depth review of HCI research methods to Lazar et al. (2010). Many of the concepts involving humans and machines in the loop explored in this thesis are inspired by ideas borrowed from HCI.

Chapter 3

"Beating the AI" to Improve Robustness

Robustness is among the key ML system attributes required for widespread real-world use. Trust is an integral element of human interaction. When a person is convinced that another person or system possesses a particular capability, that is, they understand the complexities and nuances of a task and have demonstrated the ability the perform that task reliably, they have earned trust. However, trust is easier to lose than to gain. Consider, as an example, a calculator – a system that humans have grown to rely on for complex mathematical computation. Calculators are used broadly, across a wide range of critical and non-critical applications. Now, imagine a situation where a calculator, given the input 14×3, provides the output 43. Far from being the answer to life, the universe, and everything, such an unexpected result would shatter the belief or trust in the calculator's accuracy, and potentially cast doubt on the reliability of all calculators.

Calculators are, at their core, deterministic systems programmed to follow a predetermined set of rules and algorithms. Humans are not, yet we are still able to operate and interact with one another on a basis of trust. What is it that separates the nature of interactions between humans with those with similarly non-deterministic systems such as ML models? One hypothesis is that the failure modes or patterns of such systems fundamentally differ from those made by humans, particularly in ways that are both over-confident and catastrophic.

Such ideas had been explored within the machine learning community through the concept of *adversarial examples*, first within the space of computer vision (Globerson and Roweis, 2006; Goodfellow et al., 2015) and language, and with particular relevance to this work, in the context of reading comprehension (Jia and Liang, 2017). Adversarial examples are minimal and non-semantic perturbations of instances sampled from a data distribution on which an ML system is expected to perform well, but where it fails to predict or classify correctly. In the context of our earlier thought experiment, these adversarial examples take on a critical dimension of user trust. When a user encounters such a failure, particularly when they expect the system to perform correctly, it undermines any trust in that particular system and potentially all similar systems.

There were two primary approaches for addressing weaknesses in existing systems around this time; i) automatic adversarial attacks such as ADDSENT (Jia and Liang, 2017), which described a perturbation algorithm, in this case adding a distractor sentence to the end of a paragraph, but which oftentimes strayed from a natural data distribution, and ii) manually-curated datasets targeting specific aspects or capabilities, such as multi-hop (Welbl et al., 2018; Yang et al., 2018a) or numerical (Dua et al., 2019) reasoning, which often address an important but narrow set of capabilities and which we expand on in section 3.3.

An annotation approach for overcoming these challenges, seeking a balance between attack-space breadth and a close representation of a natural distribution of examples that humans might use to interact with such models, while still allowing for crowd-sourced large-scale data collection, involves using a model-in-the-loop and encouraging crowd-workers to generate examples that the model fails to predict correctly. This work provides a thorough investigation of this approach and its effects on question diversity and reasoning requirements of the datasets produced, as well as an understanding of how reading comprehension models can benefit from access to this additional data.

The material in this chapter is based on the published work titled *"Beat the AI: Investigating Adversarial Human Annotation for Reading Comprehension"* au-

thored by **Max Bartolo**, Alastair Roberts, Johannes Welbl, Sebastian Riedel and Pontus Stenetorp.

This work was published in the Transactions of the Association for Computational Linguistics (TACL), Volume 8, 2020, Pages 662 - 678 by the MIT Press, and presented at the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020) which was held entirely online rather than in the Dominican Republic as originally planned, to avoid the need for international travel and risk of further spread of COVID-19.

The three new **datasets** introduced in this work, collectively AdversarialQA, are available publicly under a CC BY-SA 4.0 license and can be downloaded from `https://adversarialqa.github.io/` or Hugging Face datasets at `https://huggingface.co/datasets/UCLNLP/adversarial_qa`.

A public **leaderboard** is also available, powered by the Dynabench platform, at `https://dynabench.org/tasks/qa`.

3.1 Overview

Innovations in annotation methodology have been a catalyst for Reading Comprehension (RC) datasets and models. One recent trend to challenge current RC models is to involve a model in the annotation process: humans create questions adversarially, such that the model fails to answer them correctly. In this work we investigate this annotation methodology and apply it in three different settings, collecting a total of 36,000 samples with progressively stronger models in the annotation loop. This allows us to explore questions such as the reproducibility of the adversarial effect, transfer from data collected with varying model-in-the-loop strengths, and generalisation to data collected without a model. We find that training on adversarially collected samples leads to strong generalisation to non-adversarially collected datasets, yet with progressive performance deterioration with increasingly stronger models-in-the-loop. Furthermore, we find that stronger models can still learn from datasets collected with substantially weaker models-in-the-loop. When trained on data collected with a BiDAF model in the loop, RoBERTa achieves 39.9F_1 on ques-

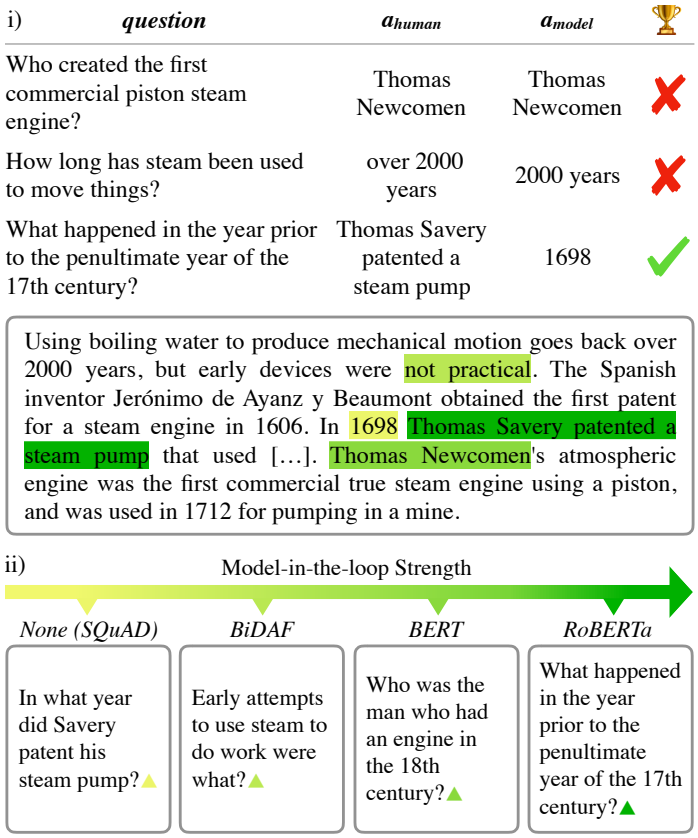


Figure 3.1: Human annotation with a model in the loop, showing: i) the "Beat the AI" annotation setting where only questions that the model does not answer correctly are accepted, and ii) questions generated this way, with a progressively stronger model in the annotation loop.

tions that it cannot answer when trained on SQuAD – only marginally lower than when trained on data collected using RoBERTa itself (41.0F_1).

3.2 Introduction

Data collection is a fundamental prerequisite for Machine Learning-based approaches to Natural Language Processing (NLP). Innovations in data acquisition methodology, such as crowdsourcing, have led to major breakthroughs in scalability and preceded the "deep learning revolution", for which they can arguably be seen as co-responsible (Deng et al., 2009; Bowman et al., 2015; Rajpurkar et al., 2016). Annotation approaches include expert annotation, for example, relying on trained linguists (Marcus et al., 1993), crowd-sourcing by non-experts (Snow et al., 2008), distant supervision (Mintz et al., 2009; Joshi et al., 2017), and leveraging

document structure (Hermann et al., 2015). The concrete data collection paradigm chosen dictates the degree of scalability, annotation cost, precise task structure (often arising as a compromise of the above) and difficulty, domain coverage, as well as resulting dataset biases and model blind spots (Jia and Liang, 2017; Schwartz et al., 2017; Gururangan et al., 2018).

A recently emerging trend in NLP dataset creation is the use of a *model-in-the-loop* when composing samples: A contemporary model is used either as a filter or directly during annotation, to identify samples wrongly predicted by the model. Examples of this method are realised in *Build It Break It, The Language Edition* (Ettinger et al., 2017), HotpotQA (Yang et al., 2018a), SWAG (Zellers et al., 2018), Mechanical Turker Descent (Yang et al., 2018b), DROP (Dua et al., 2019), CODAH (Chen et al., 2019), Quoref (Dasigi et al., 2019), and AdversarialNLI (Nie et al., 2020).[1] This approach probes model robustness and ensures that the resulting datasets pose a challenge to current models, which drives research to tackle new sets of problems.

We study this approach in the context of RC, and investigate its robustness in the face of continuously progressing models – do adversarially constructed datasets quickly become outdated in their usefulness as models grow stronger?

Based on models trained on the widely used SQuAD dataset, and following the same annotation protocol, we investigate the annotation setup where an annotator has to compose questions for which the model predicts the wrong answer. As a result, only samples that the model fails to predict correctly are retained in the dataset – see Figure 3.1 for an example.

We apply this annotation strategy with three distinct models in the loop, resulting in datasets with 12,000 samples each. We then study the reproducibility of the adversarial effect when retraining the models with the same data, as well as the generalisation ability of models trained using datasets produced with and without a model adversary. Models can, to a considerable degree, learn to generalise to more challenging questions, based on training sets collected with both

[1] The idea was alluded to at least as early as Richardson et al. (2013), but it has only recently seen wider adoption.

stronger and also weaker models in the loop. Compared to training on SQuAD, training on adversarially composed questions leads to a similar degree of generalisation to non-adversarially written questions, both for SQuAD and NaturalQuestions (Kwiatkowski et al., 2019). It furthermore leads to general improvements across the model-in-the-loop datasets we collect, as well as improvements of more than 20.0F_1 for both BERT and RoBERTa on an extractive subset of DROP (Dua et al., 2019), another adversarially composed dataset. When conducting a systematic analysis of the concrete questions different models fail to answer correctly, as well as non-adversarially composed questions, we see that the nature of the resulting questions changes: Questions composed with a model in the loop are overall more diverse, use more paraphrasing, multi-hop inference, comparisons, and background knowledge, and are generally less easily answered by matching an explicit statement that states the required information literally. Given our observations, we believe a model-in-the-loop approach to annotation shows promise and should be considered when creating future RC datasets.

To summarise, our contributions are as follows: First, an investigation into the model-in-the-loop approach to RC data collection based on three progressively stronger models, together with an empirical performance comparison when trained on datasets constructed with adversaries of different strength. Second, a comparative investigation into the nature of questions composed to be unsolvable by a sequence of progressively stronger models. Third, a study of the reproducibility of the adversarial effect and the generalisation ability of models trained in various settings.

3.3 Related Work

Constructing Challenging Datasets. Recent efforts in dataset construction have driven considerable progress in RC, yet datasets are structurally diverse and annotation methodologies vary. With its large size and combination of free-form questions with answers as extracted spans, SQuAD1.1 (Rajpurkar et al., 2016) has become an established benchmark that has inspired the construction of a se-

ries of similarly structured datasets. However, mounting evidence suggests that models can achieve strong generalisation performance merely by relying on superficial cues – such as lexical overlap, term frequencies, or entity type matching (Chen et al., 2016; Weissenborn et al., 2017; Sugawara et al., 2018). It has thus become an increasingly important consideration to construct datasets that RC models find challenging, and for which natural language understanding is a requisite for generalisation. Attempts to achieve this non-trivial aim have typically revolved around extensions to the SQuAD dataset annotation methodology. They include unanswerable questions (Trischler et al., 2017; Rajpurkar et al., 2018; Reddy et al., 2019; Choi et al., 2018), adding the option of "Yes" or "No" answers (Dua et al., 2019; Kwiatkowski et al., 2019), questions requiring reasoning over multiple sentences or documents (Welbl et al., 2018; Yang et al., 2018a), questions requiring rule interpretation or context awareness (Saeidi et al., 2018; Choi et al., 2018; Reddy et al., 2019), limiting annotator passage exposure by sourcing questions first (Kwiatkowski et al., 2019), controlling answer types by including options for dates, numbers, or spans from the question (Dua et al., 2019), as well as questions with free form answers (Nguyen et al., 2016; Kočiský et al., 2018; Reddy et al., 2019).

Adversarial Annotation. One recently adopted approach to constructing challenging datasets involves the use of an adversarial model to select examples that it does not perform well on, an approach which superficially is akin to active learning (Lewis and Gale, 1994). Here, we make a distinction between two subcategories of adversarial annotation: i) *adversarial filtering*, where the adversarial model is applied offline in a separate stage of the process, usually after data generation; examples include SWAG (Zellers et al., 2018), ReCoRD (Zhang et al., 2018), HotpotQA (Yang et al., 2018a), and HellaSWAG (Zellers et al., 2019); ii) *model-in-the-loop adversarial annotation*, where the annotator can directly interact with the adversary during the annotation process and uses the feedback to further inform the generation process; examples include CODAH (Chen et al., 2019), Quoref (Dasigi et al., 2019), DROP (Dua et al., 2019), FEVER2.0 (Thorne et al., 2019), Adversar-

ialNLI (Nie et al., 2020), as well as work by Dinan et al. (2019), Kaushik et al. (2020), and Wallace et al. (2019b) for the Quizbowl task.

We are primarily interested in the latter category, as this feedback loop creates an environment where the annotator can probe the model directly to explore its weaknesses and formulate targeted adversarial attacks. Although Dua et al. (2019) and Dasigi et al. (2019) make use of adversarial annotations for RC, both annotation setups limit the reach of the model-in-the-loop: In DROP, primarily due to the imposition of specific answer types, and in Quoref by focusing on co-reference, which is already a known RC model weakness.

In contrast, we investigate a scenario where annotators interact with a model in its original task setting – annotators must thus explore a range of natural adversarial attacks, as opposed to filtering out "easy" samples during the annotation process.

3.4 Annotation Methodology

3.4.1 Annotation Protocol

The data annotation protocol is based on SQuAD1.1, with a model in the loop, and the additional instruction that questions should only have one answer in the passage, which directly mirrors the setting in which these models were trained.

Formally, provided with a passage p, a human annotator generates a question q and selects a (human) answer a_h by highlighting the corresponding span in the passage. The input (p,q) is then given to the model, which returns a predicted (model) answer a_m. To compare the two, a word-overlap F_1 score between a_h and a_m is computed; a score above a threshold of 40% is considered a "win" for the model.[2] This process is repeated until the human "wins"; Figure 3.2 gives a schematic overview of the process. All successful (p,q,a_h) triples, that is, those which the model is unable to answer correctly, are then retained for further validation.

[2] This threshold is set after initial experiments to not be overly restrictive given acceptable answer spans, e.g., a human answer of "New York" vs. model answer "New York City" would still lead to a model "win".

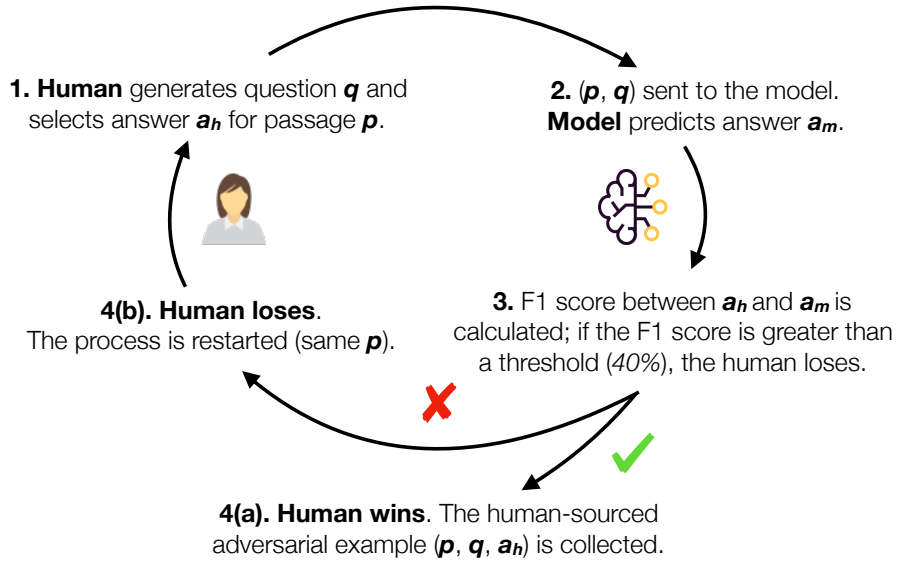


Figure 3.2: Overview of the annotation process to collect adversarially written questions from humans using a model in the loop.

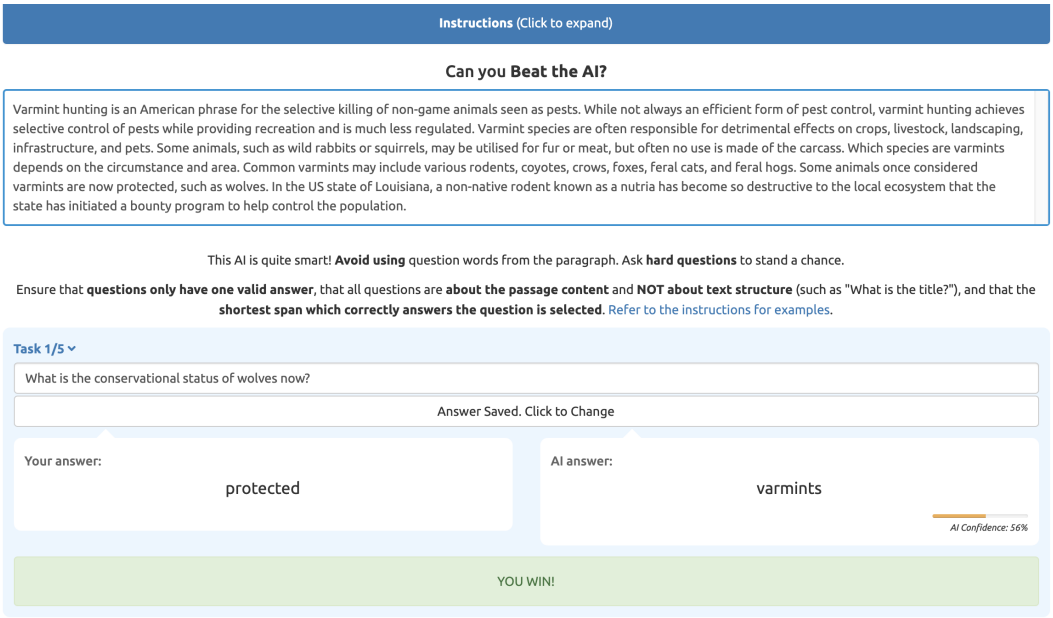


Figure 3.3: "Beat the AI" question generation interface. Human annotators are tasked with asking questions about a provided passage which the model in the loop fails to answer correctly.

3.4.2 Annotation Details

Models in the Annotation Loop. We begin by training three different models, which are used as adversaries during data annotation. As a seed dataset for training the models we select the widely used SQuAD1.1 (Rajpurkar et al., 2016) dataset,

a large-scale resource for which a variety of mature and well-performing models are readily available. Furthermore, unlike cloze-based datasets, SQuAD is robust to passage/question-only adversarial attacks (Kaushik and Lipton, 2018). We will compare dataset annotation with a series of three progressively stronger models as adversary in the loop, namely BiDAF (Seo et al., 2017), $\text{BERT}_{\text{LARGE}}$ (Devlin et al., 2019), and $\text{RoBERTa}_{\text{LARGE}}$ (Liu et al., 2019b). Each of these will serve as a model adversary in a separate annotation experiment and result in three distinct datasets; we will refer to these as $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, and $\mathcal{D}_{\text{RoBERTa}}$ respectively. Examples from the validation set of each are shown in Table 3.1. We rely on the *AllenNLP* (Gardner et al., 2018) and *Transformers* (Wolf et al., 2020) model implementations, and our models achieve EM/F_1 scores of 65.5%/77.5%, 82.7%/90.3% and 86.9%/93.6% for BiDAF, BERT, and RoBERTa respectively on the SQuAD1.1 validation set, consistent with results reported in other work.

Our choice of models reflects both the transition from LSTM-based to pre-trained transformer-based models, as well as a graduation among the latter; we investigate how this is reflected in datasets collected with each of these different models in the annotation loop. For each of the models we collect 10,000 training, 1,000 validation, and 1,000 test examples. Dataset sizes are motivated by the data efficiency of transformer-based pretrained models (Devlin et al., 2019; Liu et al., 2019b), which has improved the viability of smaller-scale data collection efforts for investigative and analysis purposes.

To ensure the experimental integrity provided by reporting all results on a held-out test set, we split the existing SQuAD1.1 validation set in half (stratified by document title) as the official test set is not publicly available. We maintain passage consistency across the training, validation and test sets of all datasets to enable like-for-like comparisons. Lastly, we use the majority vote answer as ground truth for SQuAD1.1 to ensure that all our datasets have one valid answer per question, enabling us to fairly draw direct comparisons. For clarity, we will hereafter refer to this modified version of SQuAD1.1 as $\mathcal{D}_{\text{SQuAD}}$.

BiDAF	**Passage:** *[. . .] the United Methodist Church has placed great emphasis on the importance of education. As such, the United Methodist Church established and is affiliated with around one hundred colleges [. . .] of Methodist-related Schools, Colleges, and Universities. The church operates three hundred sixty schools and institutions overseas.* **Question:** *The United Methodist Church has how many schools internationally?*
BiDAF	**Passage:** *In a purely capitalist mode of production (i.e. where professional and labor organizations cannot limit the number of workers) the workers wages will not be controlled by these organizations, or by the employer, but rather by the market. Wages work in the same way as prices for any other good. Thus, wages can be considered as a [. . .]* **Question:** *What determines worker wages?*
BiDAF	**Passage:** *[. . .] released to the atmosphere, and a separate source of water feeding the boiler is supplied. Normally water is the fluid of choice due to its favourable properties, such as non-toxic and unreactive chemistry, abundance, low cost, and its thermodynamic properties. Mercury is the working fluid in the mercury vapor turbine [. . .]* **Question:** *What is the most popular type of fluid?*
BERT	**Passage:** *[. . .] Jochi was secretly poisoned by an order from Genghis Khan. Rashid al-Din reports that the great Khan sent for his sons in the spring of 1223, and while his brothers heeded the order, Jochi remained in Khorasan. Juzjani suggests that the disagreement arose from a quarrel between Jochi and his brothers in the siege of Urgench [. . .]* **Question:** *Who went to Khan after his order in 1223?*
BERT	**Passage:** *In the Sandgate area, to the east of the city and beside the river, resided the close-knit community of keelmen and their families. They were so called because [. . .] transfer coal from the river banks to the waiting colliers, for export to London and elsewhere. In the 1630s about 7,000 out of 20,000 inhabitants of Newcastle died of plague [. . .]* **Question:** *Where did almost half the people die?*
BERT	**Passage:** *[. . .] was important to reduce the weight of coal carried. Steam engines remained the dominant source of power until the early 20th century, when advances in the design of electric motors and internal combustion engines gradually resulted in the replacement of reciprocating (piston) steam engines, with shipping in the 20th-century [. . .]* **Question:** *Why did steam engines become obsolete?*
RoBERTa	**Passage:** *[. . .] and seven other hymns were published in the Achtliederbuch, the first Lutheran hymnal. In 1524 Luther developed his original four-stanza psalm paraphrase into a five-stanza Reformation hymn that developed the theme of "grace alone" more fully. Because it expressed essential Reformation doctrine, this expanded version of "Aus [. . .]* **Question:** *Luther's reformed hymn did not feature stanzas of what quantity?*
RoBERTa	**Passage:** *[. . .] tight end Greg Olsen, who caught a career-high 77 passes for 1,104 yards and seven touchdowns, and wide receiver Ted Ginn, Jr., who caught 44 passes for 739 yards and 10 touchdowns; [. . .] receivers included veteran Jerricho Cotchery (39 receptions for 485 yards), rookie Devin Funchess (31 receptions for 473 yards and [. . .]* **Question:** *Who caught the second most passes?*
RoBERTa	**Passage:** *Other prominent alumni include anthropologists David Graeber and Donald Johanson, who is best known for discovering the fossil of a female hominid australopithecine known as "Lucy" in the Afar Triangle region, psychologist John B. Watson, American psychologist who established the psychological school of behaviorism, communication theorist Harold Innis, chess grandmaster Samuel Reshevsky, and conservative international relations scholar and White House Coordinator of Security Planning for the National Security Council Samuel P. Huntington.* **Question:** *Who thinks three moves ahead?*

Table 3.1: Validation set examples of questions collected using different RC models (BiDAF, BERT, and RoBERTa) in the annotation loop. The answer to the question is highlighted in the passage.

Crowdsourcing. We use custom-designed Human Intelligence Tasks (HITs) served through Amazon Mechanical Turk (AMT) for all annotation efforts (see Appendix A.2). Workers are required to be based in Canada, the UK, or the US, have a HIT Approval Rate greater than 98%, and have previously completed at least 1,000 HITs successfully. We experiment with and without the AMT *Master* requirement and find no substantial difference in quality, but observe a throughput reduction of

nearly 90%. We pay USD 2.00 for every question generation HIT, during which workers are required to compose up to five questions that "beat" the model in the loop (cf. Figure 3.3). The mean HIT completion times for BiDAF, BERT, and RoBERTa are 551.8s, 722.4s, and 686.4s. Furthermore we find that human workers are able to generate questions that successfully "beat" the model in the loop 59.4% of the time for BiDAF, 47.1% for BERT, and 44.0% for RoBERTa. These metrics broadly reflect the relative strength of the models.

3.4.3 Quality Control

Training and Qualification. We provide a two-part worker training interface in order to i) familiarise workers with the process, and ii) conduct a first screening based on worker outputs. The interface familiarises workers with formulating questions, and answering them through span selection. Workers are asked to generate questions for two given answers, to highlight answers for two given questions, to generate one full question-answer pair, and finally to complete a question generation HIT with BiDAF as the model in the loop. Each worker's output is then reviewed manually (by the authors); those who pass the screening are added to the pool of qualified annotators.

Manual Worker Validation. In the second annotation stage, qualified workers produce data for the "Beat the AI" question generation task. A sample of every worker's HITs is manually reviewed based on their total number of completed tasks n, determined by $\lfloor 5 \cdot \log_{10}(n) + 1 \rfloor$, chosen for convenience. This is done after every annotation batch; if workers fall below an 80% success threshold at any point, their qualification is revoked and their work is discarded in its entirety.

Question Answerability. As the models used in the annotation task become stronger, the resulting questions tend to become more complex. However, this also means that it becomes more challenging to disentangle measures of dataset quality from inherent question difficulty. As such, we use the condition of human answerability for an annotated question-answer pair as follows: It is answerable if at least one of three additional non-expert human validators can provide an answer matching the original. We conduct answerability checks on both the validation and test

	Dev		*Test*	
Resource	*EM*	F_1	*EM*	F_1
$\mathcal{D}_{\text{BiDAF}}$	63.0	76.9	62.6	78.5
$\mathcal{D}_{\text{BERT}}$	59.2	74.3	63.9	76.9
$\mathcal{D}_{\text{RoBERTa}}$	58.1	72.0	58.7	73.7

Table 3.2: Non-expert human performance for a randomly-selected validator per question.

sets, and achieve answerability scores of 87.95%, 85.41%, and 82.63% for $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, and $\mathcal{D}_{\text{RoBERTa}}$. We discard all questions deemed unanswerable from the validation and test sets, and further discard all data from any workers with less than half of their questions considered answerable. It should be emphasised that the main purpose of this process is to create a level playing field for comparison across datasets constructed for different model adversaries, and can inevitably result in valid questions being discarded. The total cost for training and qualification, dataset construction, and validation is approximately USD 27,000.

Human Performance. We select a randomly chosen validator's answer to each question and compute Exact Match (EM) and word overlap F_1 scores with the original to calculate non-expert human performance; Table 3.2 shows the result. We observe a clear trend: the stronger the model in the loop used to construct the dataset, the harder the resulting questions become for humans.

3.4.4 Dataset Statistics

Table 3.3 provides general details on the number of passages and question-answer pairs used in the different dataset splits. The average number of words in questions and answers, as well as the average longest n-gram overlap between passage and question are given in Table 3.4.

We can again observe two clear trends: From weaker towards stronger models used in the annotation loop, the average length of answers increases, and the largest n-gram overlap drops from 3 to 2 tokens. That is, on average there is a trigram overlap between the passage and question for $\mathcal{D}_{\text{SQuAD}}$, but only a bigram overlap

	#Passages			#QAs		
Resource	*Train*	*Dev*	*Test*	*Train*	*Dev*	*Test*
$\mathcal{D}_{\text{SQuAD}}$	18,891	971	1,096	87,599	5,278	5,292
$\mathcal{D}_{\text{BiDAF}}$	2,523	278	277	10,000	1,000	1,000
$\mathcal{D}_{\text{BERT}}$	2,444	283	292	10,000	1,000	1,000
$\mathcal{D}_{\text{RoBERTa}}$	2,552	341	333	10,000	1,000	1,000

Table 3.3: Number of passages and question-answer pairs for each data resource.

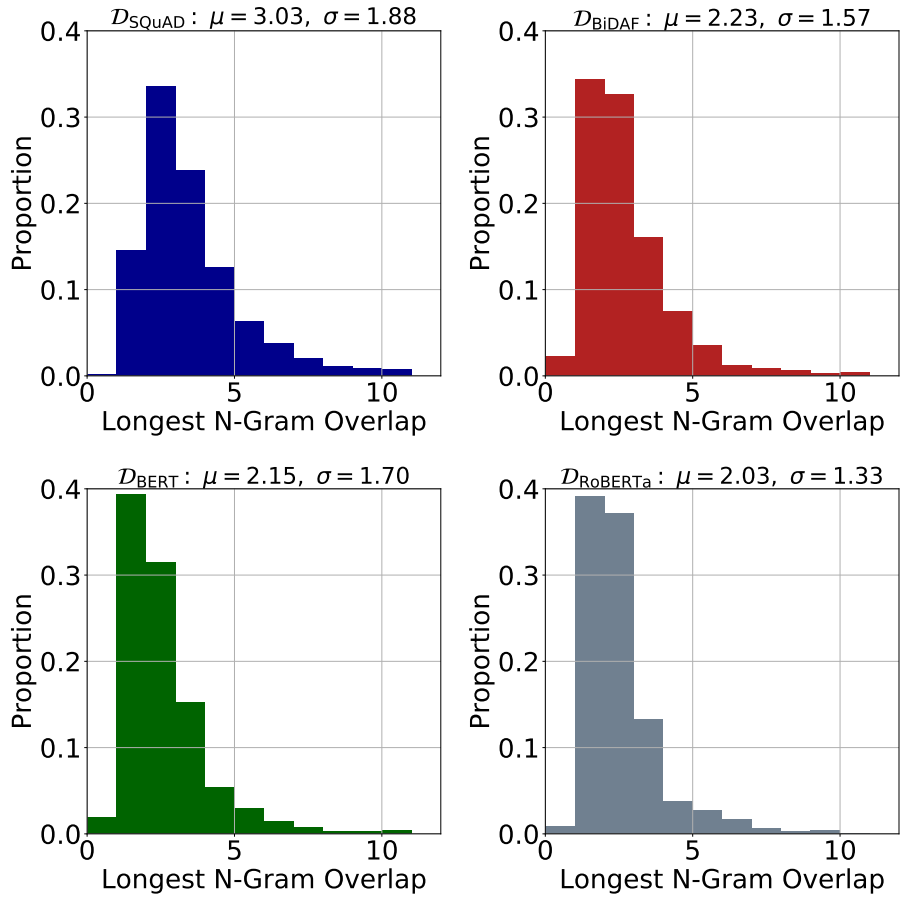


Figure 3.4: Distribution of longest n-gram overlap between passage and question for different datasets. μ: mean; σ: standard deviation.

for $\mathcal{D}_{\text{RoBERTa}}$ (Figure 3.4).[3] This is in line with prior observations on lexical overlap as a predictive cue in SQuAD (Weissenborn et al., 2017; Min et al., 2018); questions with less overlap are harder to answer for any of the three models.

[3]Note that the original SQuAD1.1 dataset can be considered a limit case of the adversarial annotation framework, in which the model in the loop always predicts the wrong answer, thus every question is accepted.

	$\mathcal{D}_{\text{SQuAD}}$	$\mathcal{D}_{\text{BiDAF}}$	$\mathcal{D}_{\text{BERT}}$	$\mathcal{D}_{\text{RoBERTa}}$
Passage length	118.1	115.3	114.6	114.6
Question length	10.3	9.8	9.8	10.0
Answer length	2.6	2.9	3.0	3.2
N-Gram overlap	3.0	2.2	2.1	2.0

Table 3.4: Average number of words per question and answer, and average longest n-gram overlap between passage and question.

We furthermore analyse question types based on the question *wh*-word. We find that – in contrast to $\mathcal{D}_{\text{SQuAD}}$ – the datasets collected with a model in the annotation loop have fewer *when*, *how* and *in* questions, and more *which*, *where* and *why* questions, as well as questions in the *other* category, which indicates increased question diversity. In terms of answer types, we observe more common noun and verb phrase clauses than in $\mathcal{D}_{\text{SQuAD}}$, as well as fewer dates, names, and numeric answers. This reflects on the strong answer-type matching capabilities of contemporary RC models. For further dataset statistics on this, see Appendix A.1. The training and validation sets used in this analysis ($\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$ and $\mathcal{D}_{\text{RoBERTa}}$) have been publicly released and are available at https://adversarialqa.github.io/.

		Original		*Re-initialised*	
Model	**Resource**	*EM*	F_1	*EM*	F_1
BiDAF	$\mathcal{D}_{\text{BiDAF}}{}^{dev}$	0.0	5.3	$10.7_{0.8}$	$20.4_{1.0}$
BERT	$\mathcal{D}_{\text{BERT}}{}^{dev}$	0.0	4.9	$19.7_{1.0}$	$30.1_{1.2}$
RoBERTa	$\mathcal{D}_{\text{RoBERTa}}{}^{dev}$	0.0	6.1	$15.7_{0.9}$	$25.8_{1.2}$
BiDAF	$\mathcal{D}_{\text{BiDAF}}{}^{test}$	0.0	5.5	$11.6_{1.0}$	$21.3_{1.2}$
BERT	$\mathcal{D}_{\text{BERT}}{}^{test}$	0.0	5.3	$18.9_{1.2}$	$29.4_{1.1}$
RoBERTa	$\mathcal{D}_{\text{RoBERTa}}{}^{test}$	0.0	5.9	$16.1_{0.8}$	$26.7_{0.9}$

Table 3.5: Consistency of the adversarial effect (or lack thereof) when retraining the models in the loop on the same data again, but with different random seeds. We report the mean and standard deviation (subscript) over 10 re-initialisation runs.

3.5 Experiments

3.5.1 Consistency of the Model in the Loop

We begin with an experiment regarding the consistency of the adversarial nature of the models in the annotation loop. Our annotation pipeline is designed to reject all samples where the model correctly predicts the answer. How reproducible is this when retraining the model with the same training data? To measure this, we evaluate the performance of instances of BiDAF, BERT, and RoBERTa, which only differ from the model used during annotation in their random initialisation and order of mini-batch samples during training. These results are shown in Table 3.5.

First, we observe – as expected given our annotation constraints – that model performance is 0.0EM on datasets created with the same respective model in the annotation loop. We observe however that retrained models do not reliably perform as poorly on those samples. For example, BERT reaches 19.7EM, whereas the original model used during annotation provides no correct answer with 0.0EM. This demonstrates that random model components can substantially affect the adversarial annotation process. The evaluation furthermore serves as a baseline for subsequent model evaluations: This much of the performance range can be learnt merely by retraining the same model. A possible takeaway for employing the model-in-the-loop annotation strategy in the future is to rely on ensembles of adversaries and reduce the dependency on one particular model instantiation, as investigated by Grefenstette et al. (2018).

3.5.2 Adversarial Generalisation

A potential problem with the focus on challenging questions is that they might be very distinct from one another, leading to difficulties in learning to generalise to and from them. We conduct a series of experiments in which we train on $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, and $\mathcal{D}_{\text{RoBERTa}}$, and observe how well models can learn to generalise to the respective test portions of these datasets. Table 3.6 shows the results, and there is a multitude of observations.

First, one clear trend we observe across all training data setups is a nega-

Model	Trained On	Evaluation (Test) Dataset											
		$\mathcal{D}_{\text{SQuAD}}$		$\mathcal{D}_{\text{BiDAF}}$		$\mathcal{D}_{\text{BERT}}$		$\mathcal{D}_{\text{RoBERTa}}$		$\mathcal{D}_{\text{DROP}}$		$\mathcal{D}_{\text{NQ}}$	
		EM	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1
BiDAF	$\mathcal{D}_{\text{SQuAD(10K)}}$	$\underline{40.9}_{0.6}$	$\underline{54.3}_{0.6}$	$7.1_{0.6}$	$\underline{15.7}_{0.6}$	$5.6_{0.3}$	$13.5_{0.4}$	$5.7_{0.4}$	$13.5_{0.4}$	$3.8_{0.4}$	$8.6_{0.6}$	$\underline{25.1}_{1.1}$	$\underline{38.7}_{0.7}$
	$\mathcal{D}_{\text{BiDAF}}$	$11.5_{0.4}$	$20.9_{0.4}$	$5.3_{0.4}$	$11.6_{0.5}$	$7.1_{0.4}$	$14.8_{0.6}$	$6.8_{0.5}$	$13.5_{0.6}$	$6.5_{0.5}$	$12.4_{0.4}$	$15.7_{1.1}$	$28.7_{0.8}$
	$\mathcal{D}_{\text{BERT}}$	$10.8_{0.3}$	$19.8_{0.4}$	$\underline{7.2}_{0.5}$	$14.4_{0.6}$	$6.9_{0.3}$	$14.5_{0.4}$	$8.1_{0.4}$	$15.0_{0.6}$	$7.8_{0.9}$	$14.5_{0.9}$	$16.5_{0.6}$	$28.3_{0.9}$
	$\mathcal{D}_{\text{RoBERTa}}$	$10.7_{0.2}$	$20.2_{0.3}$	$6.3_{0.7}$	$13.5_{0.8}$	$\underline{9.4}_{0.6}$	$\underline{17.0}_{0.6}$	$\underline{8.9}_{0.9}$	$\underline{16.0}_{0.8}$	$\underline{15.3}_{0.8}$	$\underline{22.9}_{0.8}$	$13.4_{0.9}$	$27.1_{1.2}$
BERT	$\mathcal{D}_{\text{SQuAD(10K)}}$	$\underline{69.4}_{0.5}$	$\underline{82.7}_{0.4}$	$35.1_{1.9}$	$49.3_{2.2}$	$15.6_{2.0}$	$27.3_{2.1}$	$11.9_{1.5}$	$23.0_{1.4}$	$18.9_{2.3}$	$28.9_{3.2}$	$52.9_{1.0}$	$68.2_{1.0}$
	$\mathcal{D}_{\text{BiDAF}}$	$66.5_{0.7}$	$80.6_{0.6}$	$\underline{46.2}_{1.2}$	$\underline{61.1}_{1.2}$	$\underline{37.8}_{1.4}$	$\underline{48.8}_{1.5}$	$\underline{30.6}_{0.8}$	$\underline{42.5}_{0.6}$	$\underline{41.1}_{2.3}$	$\underline{50.6}_{2.0}$	$\underline{54.2}_{1.2}$	$\underline{69.8}_{0.9}$
	$\mathcal{D}_{\text{BERT}}$	$61.2_{1.8}$	$75.7_{1.6}$	$42.9_{1.9}$	$57.5_{1.8}$	$37.4_{2.1}$	$47.9_{2.0}$	$29.3_{2.1}$	$40.0_{2.3}$	$39.4_{2.2}$	$47.6_{2.2}$	$49.9_{2.3}$	$65.7_{2.3}$
	$\mathcal{D}_{\text{RoBERTa}}$	$57.0_{1.7}$	$71.7_{1.8}$	$37.0_{2.3}$	$52.0_{2.5}$	$34.8_{1.5}$	$45.9_{2.0}$	$30.5_{2.2}$	$41.2_{2.2}$	$39.0_{3.1}$	$47.4_{2.8}$	$45.8_{2.4}$	$62.4_{2.5}$
RoBERTa	$\mathcal{D}_{\text{SQuAD(10K)}}$	$\underline{68.6}_{0.5}$	$\underline{82.8}_{0.3}$	$37.7_{1.1}$	$53.8_{1.1}$	$20.8_{1.2}$	$34.0_{1.0}$	$11.0_{0.8}$	$22.1_{0.9}$	$25.0_{2.2}$	$39.4_{2.4}$	$43.9_{3.8}$	$62.8_{3.1}$
	$\mathcal{D}_{\text{BiDAF}}$	$64.8_{0.7}$	$80.0_{0.4}$	$\underline{48.0}_{1.2}$	$\underline{64.3}_{1.1}$	$\underline{40.0}_{1.5}$	$\underline{51.5}_{1.3}$	$29.0_{1.9}$	$39.9_{1.8}$	$\underline{44.5}_{2.1}$	$\underline{55.4}_{1.9}$	$\underline{48.4}_{1.1}$	$\underline{66.9}_{0.8}$
	$\mathcal{D}_{\text{BERT}}$	$59.5_{1.0}$	$75.1_{0.9}$	$45.4_{1.5}$	$60.7_{1.5}$	$38.4_{1.8}$	$49.8_{1.7}$	$28.2_{1.5}$	$38.8_{1.5}$	$42.2_{2.3}$	$52.6_{2.0}$	$45.8_{1.1}$	$63.6_{1.1}$
	$\mathcal{D}_{\text{RoBERTa}}$	$56.2_{0.7}$	$72.1_{0.7}$	$41.4_{0.8}$	$57.1_{0.8}$	$38.4_{1.1}$	$49.5_{0.9}$	$\underline{30.2}_{1.3}$	$\underline{41.0}_{1.2}$	$41.2_{0.9}$	$51.2_{0.8}$	$43.6_{1.1}$	$61.6_{0.9}$

Table 3.6: Training models on various datasets, each with 10,000 samples, and measuring their generalisation to different evaluation datasets. Results underlined indicate the best result per model. We report the mean and standard deviation (subscript) over 10 runs with different random seeds.

tive performance progression when evaluated against datasets constructed with a stronger model in the loop. This trend holds true for all but the BiDAF model, in each of the training configurations, and for each of the evaluation datasets. For example, RoBERTa trained on $\mathcal{D}_{\text{RoBERTa}}$ achieves 72.1, 57.1, 49.5, and 41.0F_1 when evaluated on $\mathcal{D}_{\text{SQuAD}}$, $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, and $\mathcal{D}_{\text{RoBERTa}}$ respectively.

Second, we observe that the BiDAF model is not able to generalise well to datasets constructed with a model in the loop, independent of its training setup. In particular it is unable to learn from $\mathcal{D}_{\text{BiDAF}}$, thus failing to overcome some of its own blind spots through adversarial training. Irrespective of the training dataset, BiDAF consistently performs poorly on the adversarially collected evaluation datasets, and we also note a substantial performance drop when trained on $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, or $\mathcal{D}_{\text{RoBERTa}}$ and evaluated on $\mathcal{D}_{\text{SQuAD}}$.

In contrast, BERT and RoBERTa are able to partially overcome their blind spots through training on data collected with a model in the loop, and to a degree that far exceeds what would be expected from random retraining (cf. Table 3.5). For example, BERT reaches 47.9F_1 when trained and evaluated on $\mathcal{D}_{\text{BERT}}$, while RoBERTa trained on $\mathcal{D}_{\text{RoBERTa}}$ reaches 41.0F_1 on $\mathcal{D}_{\text{RoBERTa}}$, both considerably better than random retraining, or when training on the non-adversarially collected

$\mathcal{D}_{\text{SQuAD(10K)}}$ showing gains of 20.6F_1 for BERT and 18.9F_1 for RoBERTa. These observations suggest that there exists learnable structure among harder questions that can be picked up by some of the models, yet not all, as BiDAF fails to achieve this. The fact that even BERT can learn to generalise to $\mathcal{D}_{\text{RoBERTa}}$, but not BiDAF to $\mathcal{D}_{\text{BERT}}$ suggests the existence of an inherent limitation to what BiDAF can learn from these new samples, compared to BERT and RoBERTa.

More generally, we observe that training on $\mathcal{D}_{\text{S}}$, where S is a stronger RC model, helps generalise to $\mathcal{D}_{\text{W}}$, where W is a weaker model, for example, training on $\mathcal{D}_{\text{RoBERTa}}$ and testing on $\mathcal{D}_{\text{BERT}}$. On the other hand, training on $\mathcal{D}_{\text{W}}$ also leads to generalisation towards $\mathcal{D}_{\text{S}}$. For example, RoBERTa trained on 10,000 SQuAD samples reaches 22.1F_1 on $\mathcal{D}_{\text{RoBERTa}}$ ($\mathcal{D}_{\text{S}}$), whereas training RoBERTa on $\mathcal{D}_{\text{BiDAF}}$ and $\mathcal{D}_{\text{BERT}}$ ($\mathcal{D}_{\text{W}}$) bumps this number to 39.9F_1 and 38.8F_1, respectively.

Third, we observe similar performance degradation patterns for both BERT and RoBERTa on $\mathcal{D}_{\text{SQuAD}}$ when trained on data collected with increasingly stronger models in the loop. For example, RoBERTa evaluated on $\mathcal{D}_{\text{SQuAD}}$ achieves 82.8, 80.0, 75.1, and 72.1F_1 when trained on $\mathcal{D}_{\text{SQuAD(10K)}}$, $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, and $\mathcal{D}_{\text{RoBERTa}}$ respectively. This may indicate a gradual shift in the distributions of composed questions as the model in the loop gets stronger.

These observations suggest an encouraging takeaway for the model-in-the-loop annotation paradigm: Even though a particular model might be chosen as an adversary in the annotation loop, which at some point falls behind more recent state-of-the-art models, these future models can still benefit from data collected with the weaker model, and also generalise better to samples composed with the stronger model in the loop.

We further show experimental results for the same models and training datasets, but now including SQuAD as additional training data in Table 3.7. In this training setup we generally see improved generalisation to $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, and $\mathcal{D}_{\text{RoBERTa}}$. Interestingly, the relative differences between $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, and $\mathcal{D}_{\text{RoBERTa}}$ as training sets used in conjunction with SQuAD are much diminished, and especially $\mathcal{D}_{\text{RoBERTa}}$ as (part of) the training set now generalises substantially

Model	Training Dataset	Evaluation (Test) Dataset							
		$\mathcal{D}_{\text{SQuAD}}$		$\mathcal{D}_{\text{BiDAF}}$		$\mathcal{D}_{\text{BERT}}$		$\mathcal{D}_{\text{RoBERTa}}$	
		EM	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1
BiDAF	$\mathcal{D}_{\text{SQuAD}}$	$\underline{56.7}_{0.5}$	$\underline{70.1}_{0.3}$	$11.6_{1.0}$	$21.3_{1.1}$	$8.6_{0.6}$	$17.3_{0.8}$	$8.3_{0.7}$	$16.8_{0.5}$
	$\mathcal{D}_{\text{SQuAD}}$ + $\mathcal{D}_{\text{BiDAF}}$	$56.3_{0.6}$	$69.7_{0.4}$	$14.4_{0.9}$	$24.4_{0.9}$	$15.6_{1.1}$	$24.7_{1.1}$	$14.3_{0.5}$	$23.3_{0.7}$
	$\mathcal{D}_{\text{SQuAD}}$ + $\mathcal{D}_{\text{BERT}}$	$56.2_{0.6}$	$69.4_{0.6}$	$14.4_{0.7}$	$24.2_{0.8}$	$15.7_{0.6}$	$25.1_{0.6}$	$13.9_{0.8}$	$22.7_{0.8}$
	$\mathcal{D}_{\text{SQuAD}}$ + $\mathcal{D}_{\text{RoBERTa}}$	$56.2_{0.7}$	$69.6_{0.6}$	$\underline{14.7}_{0.9}$	$\underline{24.8}_{0.8}$	$\underline{17.9}_{0.5}$	$\underline{26.7}_{0.6}$	$\underline{16.7}_{1.1}$	$\underline{25.0}_{0.8}$
BERT	$\mathcal{D}_{\text{SQuAD}}$	$74.8_{0.3}$	$86.9_{0.2}$	$46.4_{0.7}$	$60.5_{0.8}$	$24.4_{1.2}$	$35.9_{1.1}$	$17.3_{0.7}$	$28.9_{0.9}$
	$\mathcal{D}_{\text{SQuAD}}$ + $\mathcal{D}_{\text{BiDAF}}$	$75.2_{0.4}$	$\underline{87.2}_{0.2}$	$52.4_{0.9}$	$66.5_{0.9}$	$40.9_{1.3}$	$51.2_{1.5}$	$32.9_{0.9}$	$44.1_{0.8}$
	$\mathcal{D}_{\text{SQuAD}}$ + $\mathcal{D}_{\text{BERT}}$	$75.1_{0.3}$	$87.1_{0.3}$	$\underline{54.1}_{1.0}$	$\underline{68.0}_{0.8}$	$43.7_{1.1}$	$54.1_{1.3}$	$34.7_{0.7}$	$45.7_{0.8}$
	$\mathcal{D}_{\text{SQuAD}}$ + $\mathcal{D}_{\text{RoBERTa}}$	$\underline{75.3}_{0.4}$	$87.1_{0.3}$	$53.0_{1.1}$	$67.1_{0.8}$	$\underline{44.1}_{1.1}$	$\underline{54.4}_{0.9}$	$\underline{36.6}_{0.8}$	$\underline{47.8}_{0.5}$
RoBERTa	$\mathcal{D}_{\text{SQuAD}}$	$73.2_{0.4}$	$86.3_{0.2}$	$48.9_{1.1}$	$64.3_{1.1}$	$31.3_{1.1}$	$43.5_{1.2}$	$16.1_{0.8}$	$26.7_{0.9}$
	$\mathcal{D}_{\text{SQuAD}}$ + $\mathcal{D}_{\text{BiDAF}}$	$\underline{73.9}_{0.4}$	$\underline{86.7}_{0.2}$	$55.0_{1.4}$	$69.7_{0.9}$	$46.5_{1.1}$	$57.3_{1.1}$	$31.9_{0.8}$	$42.4_{1.0}$
	$\mathcal{D}_{\text{SQuAD}}$ + $\mathcal{D}_{\text{BERT}}$	$73.8_{0.2}$	$\underline{86.7}_{0.2}$	$55.4_{1.0}$	$70.1_{0.9}$	$48.9_{1.0}$	$59.0_{1.2}$	$32.9_{1.3}$	$43.7_{1.4}$
	$\mathcal{D}_{\text{SQuAD}}$ + $\mathcal{D}_{\text{RoBERTa}}$	$73.5_{0.3}$	$86.5_{0.2}$	$\underline{55.9}_{0.7}$	$\underline{70.6}_{0.7}$	$\underline{49.1}_{1.2}$	$\underline{59.5}_{1.2}$	$\underline{34.7}_{1.0}$	$\underline{45.9}_{1.2}$

Table 3.7: Training models on SQuAD, as well as SQuAD combined with different adversarially created datasets. Results underlined indicate the best result per model. We report the mean and standard deviation (subscript) over 10 runs with different random seeds.

better. We see that BERT and RoBERTa both show consistent performance gains with the addition of the original SQuAD1.1 training data, but unlike in Table 3.6, this comes without any noticeable decline in performance on $\mathcal{D}_{\text{SQuAD}}$, suggesting that the adversarially constructed datasets expose inherent model weaknesses, as investigated by Liu et al. (2019a).

Furthermore, RoBERTa achieves the strongest results on the adversarially collected evaluation sets, in particular when trained on $\mathcal{D}_{\text{SQuAD}}$ + $\mathcal{D}_{\text{RoBERTa}}$. This stands in contrast to the results in Table 3.6, where training on $\mathcal{D}_{\text{BiDAF}}$ in several cases led to better generalisation than training on $\mathcal{D}_{\text{RoBERTa}}$. A possible explanation is that training on $\mathcal{D}_{\text{RoBERTa}}$ leads to a larger degree of overfitting to specific adversarial examples in $\mathcal{D}_{\text{RoBERTa}}$ than training on $\mathcal{D}_{\text{BiDAF}}$, and that the inclusion of a large number of standard SQuAD training samples can mitigate this effect.

Results for the models trained on all the datasets combined ($\mathcal{D}_{\text{SQuAD}}$, $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, and $\mathcal{D}_{\text{RoBERTa}}$) are shown in Table 3.8. These further support the previous observations and provide additional performance gains where, for example,

Model	Evaluation (Test) Dataset							
	$\mathcal{D}_{\text{SQuAD}}$		$\mathcal{D}_{\text{BiDAF}}$		$\mathcal{D}_{\text{BERT}}$		$\mathcal{D}_{\text{RoBERTa}}$	
	EM	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1
BiDAF	$57.1_{0.4}$	$70.4_{0.3}$	$17.1_{0.8}$	$27.0_{0.9}$	$20.0_{1.0}$	$29.2_{0.8}$	$18.3_{0.6}$	$27.4_{0.7}$
BERT	$\underline{75.5}_{0.2}$	$\underline{87.2}_{0.2}$	$57.7_{1.0}$	$71.0_{1.1}$	$52.1_{0.7}$	$62.2_{0.7}$	$\underline{43.0}_{1.1}$	$\underline{54.2}_{1.0}$
RoBERTa	$74.2_{0.3}$	$86.9_{0.3}$	$\underline{59.8}_{0.5}$	$\underline{74.1}_{0.6}$	$\underline{55.1}_{0.6}$	$\underline{65.1}_{0.7}$	$41.6_{1.0}$	$52.7_{1.0}$

Table 3.8: Training models on SQuAD combined with all the adversarially created datasets $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, and $\mathcal{D}_{\text{RoBERTa}}$. Results underlined indicate the best result per model. We report the mean and standard deviation (subscript) over 10 runs with different random seeds.

RoBERTa achieves F_1 scores of 86.9 on $\mathcal{D}_{\text{SQuAD}}$, 74.1 on $\mathcal{D}_{\text{BiDAF}}$, 65.1 on $\mathcal{D}_{\text{BERT}}$, and 52.7 on $\mathcal{D}_{\text{RoBERTa}}$, surpassing the best previous performance on all adversarial datasets.

Finally, we identify a risk of datasets constructed with weaker models in the loop becoming outdated. For example, RoBERTa achieves 58.2EM/73.2F_1 on $\mathcal{D}_{\text{BiDAF}}$, in contrast to 0.0EM/5.5F_1 for BiDAF – which is not far from the non-expert human performance of 62.6EM/78.5F_1 (cf. Table 3.2).

It is also interesting to note that, even when training on all the combined data (cf. Table 3.8), BERT outperforms RoBERTa on $\mathcal{D}_{\text{RoBERTa}}$ and vice versa, suggesting that there may exist weaknesses inherent to each model class.

3.5.3 Generalisation to Non-Adversarial Data

Compared to standard annotation, the model-in-the-loop approach generally results in new question distributions. Consequently, models trained on adversarially composed questions might not be able to generalise to standard ("easy") questions, thus limiting the practical usefulness of the resulting data. To what extent do models trained on model-in-the-loop questions generalise differently to standard ("easy") questions, compared to models trained on standard ("easy") questions?

To measure this we further train each of our three models on either $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, or $\mathcal{D}_{\text{RoBERTa}}$ and test on $\mathcal{D}_{\text{SQuAD}}$, with results in the $\mathcal{D}_{\text{SQuAD}}$ columns of Table 3.6. For comparison, the models are also trained on 10,000 SQuAD1.1 samples (referred to as $\mathcal{D}_{\text{SQuAD(10K)}}$) chosen from the same passages as the adversarial datasets, thus eliminating size and paragraph choice as potential confounding fac-

tors. The models are tuned for EM on the held-out $\mathcal{D}_{\text{SQuAD}}$ validation set. Note that, although performance values on the majority vote $\mathcal{D}_{\text{SQuAD}}$ dataset are lower than on the original, for the reasons described earlier, this enables direct comparisons across all datasets.

Remarkably, neither BERT nor RoBERTa show substantial drops when trained on $\mathcal{D}_{\text{BiDAF}}$ compared to training on SQuAD data (-2.1F_1, and -2.8F_1): Training these models on a dataset with a weaker model in the loop still leads to strong generalisation even to data from the original SQuAD distribution, which all models in the loop are trained on. BiDAF, on the other hand, fails to learn such information from the adversarially collected data, and drops ¿30F_1 for each of the new training sets, compared to training on SQuAD.

We also observe a gradual decrease in generalisation to SQuAD when training on $\mathcal{D}_{\text{BiDAF}}$ towards training on $\mathcal{D}_{\text{RoBERTa}}$. This suggests that the stronger the model, the more dissimilar the resulting data distribution becomes from the original SQuAD distribution. We later find further support for this explanation in a qualitative analysis (Section 3.6). It may however also be due to a limitation of BERT and RoBERTa – similar to BiDAF – in learning from a data distribution designed to beat these models; an even stronger model might learn more from, for example, $\mathcal{D}_{\text{RoBERTa}}$.

3.5.4 Generalisation to DROP and NaturalQuestions

Finally, we investigate to what extent models can transfer skills learned on the datasets created with a model in the loop to two recently introduced datasets: DROP (Dua et al., 2019), and NaturalQuestions (Kwiatkowski et al., 2019). In this experiment we select the subsets of DROP and NaturalQuestions that align with the structural constraints of SQuAD to ensure a like-for-like analysis. Specifically, we only consider questions in DROP where the answer is a span in the passage and where there is only one candidate answer. For NaturalQuestions, we consider all non-tabular long answers as passages, remove HTML tags and use the short answer as the extracted span. We apply this filtering on the validation sets for both datasets. Next we split them, stratifying by document (as we did for $\mathcal{D}_{\text{SQuAD}}$), which results

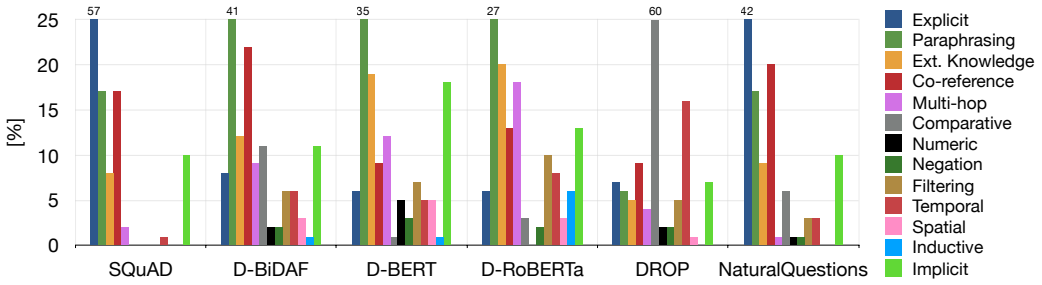


Figure 3.5: Comparison of comprehension types for the questions in different datasets. The label types are neither mutually exclusive nor comprehensive. Values above columns indicate excess of the axis range.

in 1409/1418 validation and test set examples for DROP, and 964/982 for NaturalQuestions, respectively. We denote these datasets as $\mathcal{D}_{\text{DROP}}$ and $\mathcal{D}_{\text{NQ}}$ for clarity and distinction from their unfiltered versions. We consider the same models and training datasets as before, but tune on the respective validation sets of $\mathcal{D}_{\text{DROP}}$ and $\mathcal{D}_{\text{NQ}}$. Table 3.6 shows the results of these experiments in the respective $\mathcal{D}_{\text{DROP}}$ and $\mathcal{D}_{\text{NQ}}$ columns.

First, we observe clear generalisation improvements towards $\mathcal{D}_{\text{DROP}}$ across all models compared to training on $\mathcal{D}_{\text{SQuAD(10K)}}$ when training on any of $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, or $\mathcal{D}_{\text{RoBERTa}}$. That is, including a model in the loop for the training dataset leads to improved transfer towards $\mathcal{D}_{\text{DROP}}$. Note that DROP also makes use of a BiDAF model in the loop during annotation; these results are in line with our prior observations when testing the same setups on $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$ and $\mathcal{D}_{\text{RoBERTa}}$, compared to training on $\mathcal{D}_{\text{SQuAD(10K)}}$.

Second, we observe overall strong transfer results towards $\mathcal{D}_{\text{NQ}}$, with up to 69.8F_1 for a BERT model trained on $\mathcal{D}_{\text{BiDAF}}$. Note that this result is similar to, and even slightly improves over model training with SQuAD data of the same size. That is, relative to training on SQuAD data, training on adversarially collected data $\mathcal{D}_{\text{BiDAF}}$ does not impede generalisation to the $\mathcal{D}_{\text{NQ}}$ dataset, which was created without a model in the annotation loop. We then however see a similar negative performance progression as observed before when testing on $\mathcal{D}_{\text{SQuAD}}$: the stronger the model in the annotation loop of the training dataset, the lower the test accuracy on test data from a data distribution composed without a model in the loop.

3.6 Qualitative Analysis

Having applied the general model-in-the-loop methodology on models of varying strength, we next perform a qualitative comparison of the nature of the resulting questions. As reference points we also include the original SQuAD questions, as well as DROP and NaturalQuestions in this comparison: These datasets are both constructed to overcome limitations in SQuAD and have subsets sufficiently similar to SQuAD to make an analysis possible. Specifically, we seek to understand the qualitative differences in terms of reading comprehension challenges posed by the questions in each of these datasets.

3.6.1 Comprehension Requirements

There exists a variety of prior work that seeks to understand the types of knowledge, comprehension skills or types of reasoning required to answer questions based on text (Rajpurkar et al., 2016; Clark et al., 2018; Sugawara et al., 2020; Dua et al., 2019; Dasigi et al., 2019); we are however unaware of any commonly accepted formalism. We take inspiration from these but develop our own taxonomy of comprehension requirements which suits the datasets analysed, see Appendix A.3 for a detailed breakdown and examples of our annotation catalogue. Our taxonomy contains 13 labels, most of which are commonly used in other work. However, the following three deserve additional clarification: i) *explicit* – for which the answer is stated nearly word-for-word in the passage as it is in the question, ii) *filtering* – a set of answers is narrowed down to select one by some particular distinguishing feature, and iii) *implicit* – the answer builds on information implied by the passage and does not otherwise require any of the other types of reasoning.

We annotate questions with labels from this catalogue in a manner that is not mutually exclusive, and neither fully comprehensive; the development of such a catalogue is itself very challenging. Instead, we focus on capturing the most salient characteristics of each given question, and assign it up to three of the labels in our catalogue. In total, we analyse 100 samples from the validation set of each of the datasets; Figure 3.5 shows the results.

3.6.2 Observations

An initial observation is that the majority (57%) of answers to SQuAD questions are stated explicitly, without comprehension requirements beyond the literal level. This number decreases substantially for any of the model-in-the-loop datasets derived from SQuAD (e.g., ~8% for $\mathcal{D}_{\text{BiDAF}}$) and also $\mathcal{D}_{\text{DROP}}$, yet 42% of questions in $\mathcal{D}_{\text{NQ}}$ share this property. In contrast to SQuAD, the model-in-the-loop questions generally tend to involve more paraphrasing. They also require more external knowledge, and multi-hop inference (beyond co-reference resolution) with an increasing trend for stronger models used in the annotation loop. Model-in-the-loop questions further fan out into a variety of small, but non-negligible proportions of more specific types of inference required for comprehension, for example, spatial or temporal inference (both going beyond explicitly stated spatial or temporal information) – SQuAD questions rarely require these at all. Some of these more particular inference types are common features of the other two datasets, in particular *comparative* questions for DROP (60%) and to a small extent also NaturalQuestions. Interestingly, $\mathcal{D}_{\text{BiDAF}}$ possesses the largest number of comparison questions (11%) among our model-in-the-loop datasets, whereas $\mathcal{D}_{\text{BERT}}$ and $\mathcal{D}_{\text{RoBERTa}}$ only possess 1% and 3%, respectively. This offers an explanation for our previous observation in Table 3.6, where BERT and RoBERTa perform better on $\mathcal{D}_{\text{DROP}}$ when trained on $\mathcal{D}_{\text{BiDAF}}$ rather than on $\mathcal{D}_{\text{BERT}}$ or $\mathcal{D}_{\text{RoBERTa}}$. It is likely that BiDAF as a model in the loop is worse than BERT and RoBERTa at *comparative* questions, as evidenced by the results in Table 3.6 with BiDAF reaching 8.6F_1, BERT reaching 28.9F_1, and RoBERTa reaching 39.4F_1 on $\mathcal{D}_{\text{DROP}}$ (when trained on $\mathcal{D}_{\text{SQuAD(10K)}}$).

The distribution of NaturalQuestions contains elements of both the SQuAD and $\mathcal{D}_{\text{BiDAF}}$ distributions, which offers a potential explanation for the strong performance on $\mathcal{D}_{\text{NQ}}$ of models trained on $\mathcal{D}_{\text{SQuAD(10K)}}$ and $\mathcal{D}_{\text{BiDAF}}$. Finally, the gradually shifting distribution away from both SQuAD and NaturalQuestions as the model-in-the-loop strength increases reflects our prior observations on the decreasing performance on SQuAD and NaturalQuestions of models trained on datasets with progressively stronger models in the loop.

3.7 Discussion and Conclusion

We have investigated an RC annotation paradigm that requires a model in the loop to be "beaten" by an annotator. Applying this approach with progressively stronger models in the loop (BiDAF, BERT, and RoBERTa), we produced three separate datasets. Using these datasets, we investigated several questions regarding the annotation paradigm, in particular whether such datasets grow outdated as stronger models emerge, and their generalisation to standard (non-adversarially collected) questions. We found that stronger models can still learn from data collected with a weak adversary in the loop, and their generalisation improves even on datasets collected with a stronger adversary. Models trained on data collected with a model in the loop further generalise well to non-adversarially collected data, both on SQuAD and on NaturalQuestions, yet we observe a gradual deterioration in performance with progressively stronger adversaries.

We see our work as a contribution towards the emerging paradigm of model-in-the-loop annotation. While this paper has focused on RC, with SQuAD as the original dataset used to train model adversaries, we see no reason in principle why findings would not be similar for other tasks using the same annotation paradigm, when crowdsourcing challenging samples with a model in the loop. We would expect the insights and benefits conveyed by model-in-the-loop annotation to be the greatest on mature datasets where models exceed human performance: Here the resulting data provides a magnifying glass on model performance, focused in particular on samples which models struggle on. On the other hand, applying the method to datasets where performance has not yet plateaued would likely result in a more similar distribution to the original data, which is challenging to models a priori. We hope that the series of experiments on replicability, observations on transfer between datasets collected using models of different strength, as well as our findings regarding generalisation to non-adversarially collected data, can support and inform future research and annotation efforts using this paradigm.

3.8 Reflection

The work presented in this chapter allowed a deep exploration into how humans interact with machines in the context of probing for failure modes in machine learning systems and working to improve robustness in those areas. In particular, some high-level takeaways that have influenced subsequent work include:

Annotators are effective at creating diverse adversarial examples. This may seem quite obvious today, but was far from being so at the time that this work was introduced. In fact, the annotation interfaces and incentive structures presented in this chapter underwent at least three full re-designs at the prototype stage with the authors of this work as initial testers. In many cases, we struggled to identify effective techniques for "beating" even the simplest system (BiDAF at that time). It was around the third or fourth iteration of the setup that collaborators and members of the group started asking questions that a robust system should easily get correct but was clearly struggling with. We rolled this version out to crowdworkers who also had considerable success.

Adversarial examples create more challenging evaluation scenarios. We find a clear negative performance progression when evaluating systems against datasets adversarially constructed with a stronger model in the loop. This suggests that as systems improve, we can use those same systems in the loop to push the limits of what is possible and what we can measure.

Training on adversarial examples substantially improves performance. We find that both BERT and RoBERTa are able to partially overcome their blind spots through adversarial training, demonstrated by the jump in performance from when they are trained on SQuAD.

Generalisation to external datasets. As mentioned in the introductory section of this chapter, one highly effective approach to targeting model weaknesses was the curation of specifically-targeted datasets such as DROP (Dua et al., 2019) targeting numerical reasoning. It is promising that a more general approach like the one discussed in this chapter also yields substantial performance improvements on such targeted external datasets.

Signs of a distributional shift. The monotonic decrease in performance demonstrated by all three of the systems explored in this work on the SQuAD validation set, when trained on examples sourced using increasingly strong models in the loop, suggests a distributional shift away from the original SQuAD setting. However, this effect is easily mitigated when combining the adversarially collected data with SQuAD training data. We find that in this setting, model performance on SQuAD improves slightly instead.

Early signs of the benefits of data scale. All models get better with more data. In fact, for all three models, training on the combined $87,000$ SQuAD examples and $30,000$ adversarially-collected examples gave a considerable performance increase across the board, including a marginal improvement on SQuAD.

Model-specific blind spots. It's also worth noting that even in the setting of training on all available data, BERT still outperforms RoBERTa on $\mathcal{D}_{\text{RoBERTa}}$ and vice versa, suggesting the existence of blind spots or failure modes specific to each model.

Among the contributions of this work was the AdversarialQA resource and collect of training and evaluation artefacts. AdversarialQA was also made available for download through the Hugging Face dataset hub, where it was consistently among one of the most downloaded Question Answering datasets (along with SQuAD and SQuAD2.0 for many years, and amassing over 1.34 million total downloads. The AdversarialQA datasets have also been used to drive real-world impact across various user-facing and enterprise applications, where they have contributed to both general performance improvements and improved robustness of question answering systems requiring comprehension capabilities.

Furthermore, the AdversarialQA datasets have been included in various dataset collections commonly used to train Large Language Models such as T0-SF (Sanh et al., 2022), one of the earliest collections of instruction-following datasets. This collection was used to train FLAN-T5 (Chung et al., 2024) which was among the earliest general instruction-following language models. In the context of Large Language Models, it is further worth noting that AdversarialQA test sets remain private, potentially posing an interesting test bed for evaluating the effects of training set

contamination or test set leakage in such systems.

Beyond the resource contributions and robustness improvements to various real-world question answering systems across domains, this research has also inspired work on dynamic adversarial data collection and benchmarking as we will discuss in the next chapter.

Chapter 4

Dynamic Adversarial Data Collection and Benchmarking

The research presented in the preceding chapter, alongside the contemporaneous work on AdversarialNLI (Nie et al., 2020) by researchers at Facebook AI Research, offered compelling evidence of the benefits of Dynamic Adversarial Data Collection (DADC). This method proved its utility in both training machine learning models more effectively and creating robust benchmarks to evaluate their performance.

During discussions to source funding for and build out a platform for adversarial data collection at scale, our group at UCL was introduced to Douwe Kiela at Facebook AI Research (FAIR) who had both provided the vision for the AdversarialNLI work and secured early funding to build out a prototype for what would become Dynabench, a large scale collaboration between Facebook AI Research, University College London, Stanford University, the University of North Carolina at Chapel Hill, and many others to drive improvements in model benchmarking and robustness by collecting human data dynamically with models in the loop.

This chapter is based on the published work titled *"Dynabench: Rethinking Benchmarking in NLP"* authored by Douwe Kiela, **Max Bartolo**, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts and Adina Williams. This work was published at the 2021 Annual Conference

of the North American Chapter of the Association for Computational Linguistics (NAACL) held online between the 6th and 11th June, 2021, due to the ongoing COVID-19 pandemic.

It is also influenced by the work titled *"Dynatask: A Framework for Creating Dynamic AI Benchmark Tasks"* authored by Tristan Thrush, Kushal Tirumala, Anmol Gupta, **Max Bartolo**, Pedro Rodriguez, Tariq Kane, William Gaviria Rojas, Peter Mattson, Adina Williams and Douwe Kiela, presented as a system demonstration at ACL 2022.

The thesis author's contributions involved early design and development of the Dynabench research platform, including writing parts of the initial codebase, running countless experiments and supporting and advising various others, as well as writing parts of the published research papers. The Dynabench effort was led primarily by Douwe Kiela along with many other contributors, and this chapter is included in this thesis for context, linking the previous chapter which provided inspiration and motivation for this work, as well as the upcoming two chapters which are built on the work presented here. The AdversarialQA datasets also formed Round 1 of the Question Answering task on Dynabench.

The Dynabench platform is currently managed by the MLCommons, where the thesis author co-chairs the Dynabench Working Group (recently merging with the Data-centric and Machine Learning Research (DMLR) Working Group) responsible for the maintenance of existing tasks and development and running of new research programmes, since January 2023.

4.1 Overview

We introduce Dynabench, an open-source platform for dynamic dataset creation and model benchmarking. Dynabench runs in a web browser and supports human-and-model-in-the-loop dataset creation: annotators seek to create examples that a target model will misclassify, but that another person will not. In this paper, we argue that Dynabench addresses a critical need in our community: contemporary models quickly achieve outstanding performance on benchmark tasks but nonetheless

fail on simple challenge examples and falter in real-world scenarios. With Dynabench, dataset creation, model development, and model assessment can directly inform each other, leading to more robust and informative benchmarks. We report on four initial NLP tasks, illustrating these concepts and highlighting the promise of the platform, and address potential objections to dynamic benchmarking as a new standard for the field.

4.2 Introduction

While it used to take decades for machine learning models to surpass estimates of human performance on benchmark tasks, that milestone is now routinely reached within just a few years for newer datasets (see Figure 4.1). As with the rest of AI, NLP has advanced rapidly thanks to improvements in computational power, as well as algorithmic breakthroughs, ranging from attention mechanisms (Bahdanau et al., 2015; Luong et al., 2015), to Transformers (Vaswani et al., 2017), to pre-trained language models (Howard and Ruder, 2018; Devlin et al., 2019; Liu et al., 2019b; Radford et al., 2019; Brown et al., 2020). Equally important has been the rise of benchmarks that support the development of ambitious new data-driven models and that encourage apples-to-apples model comparisons. Benchmarks provide a north star goal for researchers, and are part of the reason we can confidently say we have made great strides in our field.

In light of these developments, one might be forgiven for thinking that NLP has created models with human-like language capabilities. Practitioners know that, despite our progress, we are actually far from this goal. Models that achieve superhuman performance on benchmark tasks (according to the narrow criteria used to define human performance) nonetheless fail on simple challenge examples and falter in real-world scenarios. A substantial part of the problem is that our benchmark tasks are not adequate proxies for the sophisticated and wide-ranging capabilities we are targeting: they contain inadvertent and unwanted statistical and social biases that make them artificially easy and misaligned with our true goals.

We believe the time is ripe to radically rethink benchmarking. In this paper,

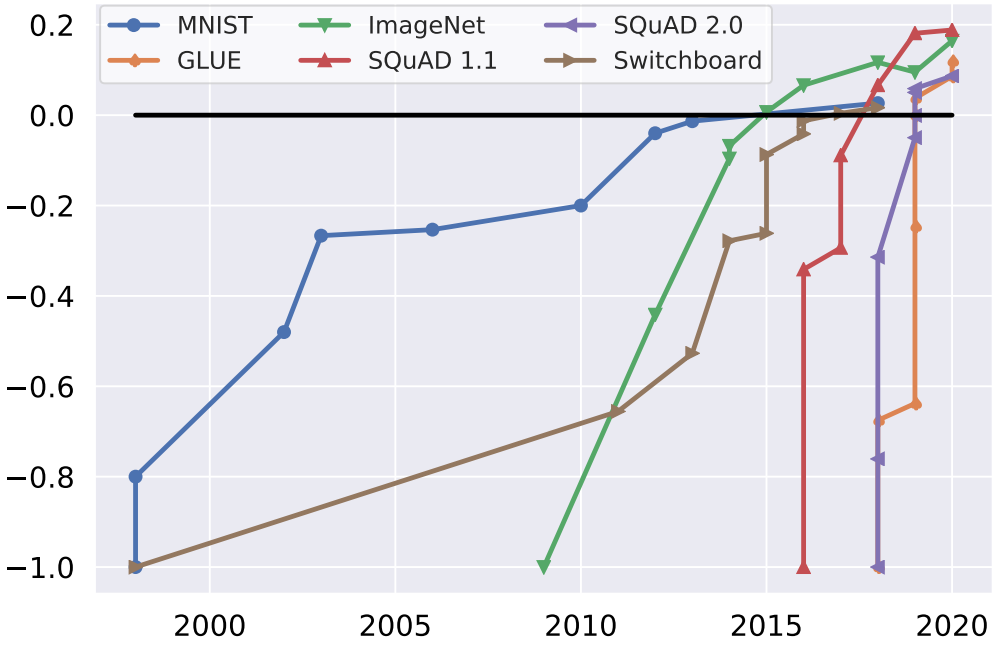


Figure 4.1: Benchmark saturation over time for popular benchmarks, normalized with initial performance at minus one and human performance at zero.

which both takes a position and seeks to offer a partial solution, we introduce Dynabench, an open-source, web-based research platform for dynamic data collection and model benchmarking. The guiding hypothesis behind Dynabench is that we can make even faster progress if we evaluate models and collect data dynamically, with humans and models in the loop, rather than the traditional static way.

Concretely, Dynabench hosts tasks for which we dynamically collect data against state-of-the-art models in the loop, over multiple rounds. The stronger the models are and the fewer weaknesses they have, the lower their error rate will be when interacting with humans, giving us a concrete metric—i.e., how well do AI systems perform when interacting with humans? This reveals the shortcomings of state-of-the-art models, and it yields valuable training and assessment data which the community can use to develop even stronger models.

In this paper, we first document the background that led us to propose this platform. We then describe the platform in technical detail, report on findings for four initial tasks, and address possible objections. We finish with a discussion of

future plans and next steps.

4.3 Background

Progress in NLP has traditionally been measured through a selection of task-level datasets that gradually became accepted benchmarks (Marcus et al., 1993; Pradhan et al., 2012). Recent well-known examples include the Stanford Sentiment Treebank (Socher et al., 2013), SQuAD (Rajpurkar et al., 2016, 2018), SNLI (Bowman et al., 2015), and MultiNLI (Williams et al., 2018). More recently, multi-task benchmarks such as SentEval (Conneau and Kiela, 2018), DecaNLP (McCann et al., 2019), GLUE (Wang et al., 2018), and SuperGLUE (Wang et al., 2019) were proposed with the aim of measuring general progress across several tasks. When the GLUE dataset was introduced, "solving GLUE" was deemed "beyond the capability of current transfer learning methods" (Wang et al., 2018). However, GLUE saturated within a year and its successor, SuperGLUE, already has models rather than humans at the top of its leaderboard. These are remarkable achievements, but there is an extensive body of evidence indicating that these models do not in fact have the human-level natural language capabilities one might be lead to believe.

4.3.1 Challenge Sets and Adversarial Settings

Whether our models have learned to solve tasks in robust and generalizable ways has been a topic of much recent interest. Challenging test sets have shown that many state-of-the-art NLP models struggle with compositionality (Nie et al., 2019; Kim and Linzen, 2020; Yu and Ettinger, 2020; White et al., 2020), and find it difficult to pass the myriad stress tests for social (Rudinger et al., 2018; May et al., 2019; Nangia et al., 2020) and/or linguistic competencies (Geiger et al., 2018; Naik et al., 2018; Glockner et al., 2018; White et al., 2018; Warstadt et al., 2019; Gauthier et al., 2020; Hossain et al., 2020; Jeretic et al., 2020; Lewis et al., 2021a; Saha et al., 2020; Schuster et al., 2020; Sugawara et al., 2020; Warstadt et al., 2020). Yet, challenge sets may suffer from performance instability (Liu et al., 2019a; Rozen et al., 2019; Zhou et al., 2020) and often lack sufficient statistical power (Card et al., 2020), suggesting that, although they may be valuable assessment tools, they

are not sufficient for ensuring that our models have achieved the learning targets we set for them.

Models are susceptible to adversarial attacks, and despite impressive task-level performance, state-of-the-art systems still struggle to learn robust representations of linguistic knowledge (Ettinger et al., 2017), as also shown by work analyzing model diagnostics (Ettinger, 2020; Ribeiro et al., 2020). For example, question answering models can be fooled by simply adding a relevant sentence to the passage (Jia and Liang, 2017).

Text classification models have been shown to be sensitive to single input character change (Ebrahimi et al., 2018b) and first-order logic inconsistencies (Minervini and Riedel, 2018). Similarly, machine translation systems have been found susceptible to character-level perturbations (Ebrahimi et al., 2018a) and synthetic and natural noise (Belinkov and Bisk, 2018; Khayrallah and Koehn, 2018). Natural language inference models can be fooled by simple syntactic heuristics or hypothesis-only biases (Gururangan et al., 2018; Poliak et al., 2018; Tsuchiya, 2018; Belinkov et al., 2019; McCoy et al., 2019). Dialogue models may ignore perturbations of dialogue history (Sankar et al., 2019). More generally, Wallace et al. (2019a) find universal adversarial perturbations forcing targeted model errors across a range of tasks. Recent work has also focused on evaluating model diagnostics through counterfactual augmentation (Kaushik et al., 2020), decision boundary analysis (Gardner et al., 2020; Swayamdipta et al., 2020), and behavioural testing (Ribeiro et al., 2020).

4.3.2 Adversarial Training and Testing

Research progress has traditionally been driven by a cyclical process of resource collection and architectural improvements. Similar to Dynabench, recent work seeks to embrace this phenomenon, addressing many of the previously mentioned issues through an iterative human-and-model-in-the-loop annotation process (Yang et al., 2018b; Dinan et al., 2019; Chen et al., 2019; Bartolo et al., 2020; Nie et al., 2020), to find "unknown unknowns" (Attenberg et al., 2015) or in a never-ending or life-long learning setting (Silver et al., 2013; Mitchell et al., 2018). The Adver-

sarial NLI (ANLI) dataset (Nie et al., 2020), for example, was collected with an adversarial setting over multiple rounds to yield "a 'moving post' dynamic target for NLU systems, rather than a static benchmark that will eventually saturate". In its few-shot learning mode, GPT-3 barely shows "signs of life" (Brown et al., 2020) (i.e., it is barely above random) on ANLI, which is evidence that we are still far away from human performance on that task.

4.3.3 Other Related Work

While crowdsourcing has been a boon for large-scale NLP dataset creation (Snow et al., 2008; Munro et al., 2010), we ultimately want NLP systems to handle "natural" data (Kwiatkowski et al., 2019) and be "ecologically valid" (de Vries et al., 2020). (Ethayarajh and Jurafsky, 2020) analyze the distinction between what leaderboards incentivize and "what is useful in practice" through the lens of microeconomics. A natural setting for exploring these ideas might be dialogue (Hancock et al., 2019; Shuster et al., 2020). Other works have pointed out misalignments between maximum-likelihood training on i.i.d. train/test splits and human language (Linzen, 2020; Stiennon et al., 2020).

We think there is widespread agreement that something has to change about our standard evaluation paradigm and that we need to explore alternatives. The persistent misalignment between benchmark performance and performance on challenge and adversarial test sets reveals that standard evaluation paradigms overstate the ability of our models to perform the tasks we have set for them. Dynabench offers one path forward from here, by allowing researchers to combine model development with the stress-testing that needs to be done to achieve true robustness and generalization.

4.4 Dynabench

Dynabench is a platform that encompasses different *tasks*. Data for each task is collected over multiple *rounds*, each starting from the current state of the art. In every round, we have one or more *target models* "in the loop." These models interact with humans, be they expert linguists or crowdworkers, who are in a position to

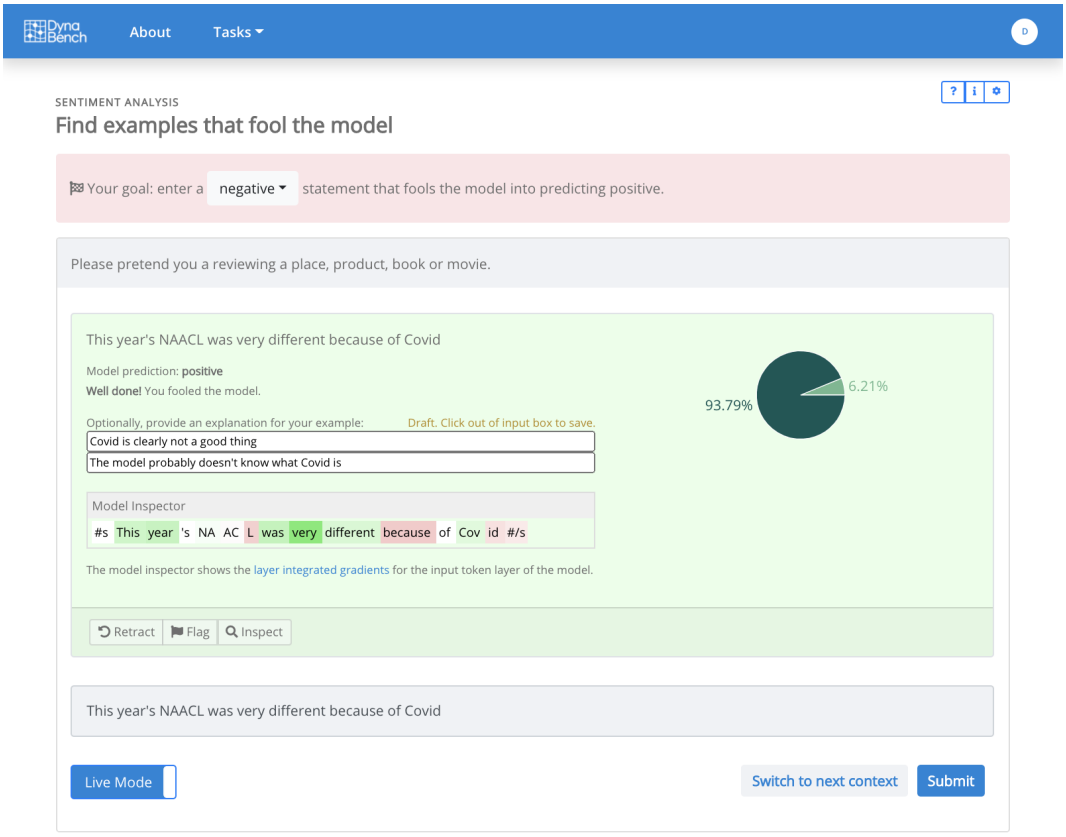


Figure 4.2: The Dynabench example creation interface for sentiment analysis with illustrative example.

identify models' shortcomings by providing *examples* for an optional *context*. Examples that models get wrong, or struggle with, can be validated by other humans to ensure their correctness. The data collected through this process can be used to evaluate state-of-the-art models, and to train even stronger ones, hopefully creating a virtuous cycle that helps drive progress in the field. Figure 4.2 provides a sense of what the example creation interface looks like.

As a large-scale collaborative effort, the platform is meant to be a platform technology for human-and-model-in-the-loop evaluation that belongs to the entire community. In the current iteration, the platform is set up for dynamic *adversarial* data collection, where humans can attempt to find model-fooling examples. This design choice is due to the fact that the *average* case, as measured by maximum likelihood training on i.i.d. datasets, is much less interesting than the *worst* (i.e., adversarial) case, which is what we want our systems to be able to handle if they

are put in critical systems where they interact with humans in real-world settings.

However, Dynabench is not limited to the adversarial setting, and one can imagine scenarios where humans are rewarded not for fooling a model or ensemble of models, but for finding examples that models, even if they are right, are very uncertain about, perhaps in an active learning setting. Similarly, the paradigm is perfectly compatible with collaborative settings that utilize human feedback, or even negotiation. The crucial aspect of this proposal is the fact that models and humans interact live "in the loop" for evaluation and data collection.

One of the aims of this platform is to put expert linguists center stage. Creating model-fooling examples is not as easy as it used to be, and finding interesting examples is rapidly becoming a less trivial task. In ANLI, the validated model error rate for crowd workers in the later rounds went below 1-in-10 (Nie et al., 2020), while in "Beat the AI", human performance decreased while time per valid adversarial example went up with stronger models in the loop (Bartolo et al., 2020). For expert linguists, we expect the model error to be much higher, but if the platform actually lives up to its virtuous cycle promise, that error rate will go down quickly. Thus, we predict that linguists with expertise in exploring the decision boundaries of machine learning models will become essential.

While we are primarily motivated by evaluating progress, both ANLI and "Beat the AI" show that models can overcome some of their existing blind spots through adversarial training. They also find that best model performance is still quite far from that of humans, suggesting that while the collected data appears to lie closer to the model decision boundaries, there still exist adversarial examples beyond the remit of current model capabilities.

4.4.1 Features and Implementation Details

Dynabench offers low-latency, real-time feedback on the behavior of state-of-the-art NLP models. The technology stack is based on PyTorch (Paszke et al., 2019), with models served via TorchServe.[1] The platform not only displays prediction probabilities, but through an "inspect model" functionality, allows the user to examine

[1] https://pytorch.org/serve

the token-level layer integrated gradients (Sundararajan et al., 2017), obtained via the Captum interpretability library.[2]

For each example, we allow the user to *explain* what the correct label is, as well as why they think it fooled a model if the model got it wrong; or why the model might have been fooled if it wasn't. All collected model-fooling (or, depending on the task, even non-model-fooling) examples are verified by other humans to ensure their validity.

Task owners can collect examples through the web interface, by engaging with the community, or through Mephisto,[3] which makes it easy to connect, e.g., Mechanical Turk workers to the exact same backend. All collected data will be open sourced, in an anonymized fashion.

In its current mode, Dynabench could be described as a fairly conservative departure from the status quo. It is being used to develop datasets that support the same metrics that drive existing benchmarks. The crucial change is that the datasets are now dynamically created, allowing for more kinds of evaluation—e.g., tracking progress through rounds and across different conditions.

4.4.2 Initial Tasks

We have selected four official tasks as a starting point, which we believe represent an appropriate cross-section of the field at this point in time. Natural Language Inference (NLI) and Question Answering (QA) are canonical tasks in the field. Sentiment analysis is a task that some consider "solved" (and is definitely treated as such, with all kinds of ethically problematic repercussions), which we show is not the case. Hate speech is very important as it can inflict harm on people, yet classifying it remains challenging for NLP.

Natural language inference. Built upon the semantic foundation of natural logic (Sánchez Valencia, 1991, i.a.) and hailing back much further (van Benthem, 2008), NLI is one of the quintessential natural language understanding tasks. NLI, also known as 'recognizing textual entailment' (Dagan et al., 2006), is often formulated

[2] https://captum.ai/
[3] https://github.com/facebookresearch/Mephisto

as a 3-way classification problem where the input is a context sentence paired with a hypothesis, and the output is a label (entailment, contradiction, or neutral) indicating the relation between the pair.

We build on the ANLI dataset (Nie et al., 2020) and its three rounds to seed the Dynabench NLI task. During the ANLI data collection process, the annotators were presented with a context (extracted from a pre-selected corpus) and a desired target label, and asked to provide a hypothesis that fools the target model adversary into misclassifying the example. If the target model is fooled, the annotator was invited to speculate about why, or motivate why their example was right. The target model of the first round (R1) was a single BERT-Large model fine-tuned on SNLI and MNLI, while the target model of the second and third rounds (R2, R3) was an ensemble of RoBERTa-Large models fine-tuned on SNLI, MNLI, FEVER (Thorne et al., 2018) recast as NLI, and all of the ANLI data collected prior to the corresponding round. The contexts for Round 1 and Round 2 were Wikipedia passages curated in Yang et al. (2018a) and the contexts for Round 3 were from various domains. Results indicate that state-of-the-art models (which can obtain 90%+ accuracy on SNLI and MNLI) cannot exceed 50% accuracy on rounds 2 and 3.

With the launch of Dynabench, we have started collection of a fourth round, which has several innovations: not only do we select candidate contexts from a more diverse set of Wikipedia featured articles but we also use an ensemble of two different models with different architectures as target adversaries to increase diversity and robustness. Moreover, the ensemble of adversaries will help mitigate issues with creating a dataset whose distribution is too closely aligned to a particular target model or architecture. Additionally, we are collecting two types of natural language explanations: why an example is correct and why a target model might be wrong. We hope that disentangling this information will yield an additional layer of interpretability and yield models that are as least as explainable as they are robust.

Question answering. The QA task takes the same format as SQuAD1.1 (Rajpurkar et al., 2016), i.e., given a context and a question, extract an answer from the context as a continuous span of text. The first round of adversarial QA (AQA) data comes

from "Beat the AI" (Bartolo et al., 2020). During annotation, crowd workers were presented with a context sourced from Wikipedia, identical to those in SQuAD1.1, and asked to write a question and select an answer. The annotated answer was compared to the model prediction using a word-overlap F_1 threshold and, if sufficiently different, considered to have fooled the model. The target models in round 1 were BiDAF (Seo et al., 2017), BERT-Large, and RoBERTa-Large.

The model in the loop for the current round is RoBERTa trained on the examples from the first round combined with SQuAD1.1. Despite the super-human performance achieved on SQuAD1.1, machine performance is still far from humans on the current leaderboard. In the current phase, we seek to collect rich and diverse examples, focusing on improving model robustness through generative data augmentation, to provide more challenging model adversaries in this constrained task setting. We should emphasize that we don't consider this task structure representative of the broader definition even of closed-domain QA, and are looking to expand this to include unanswerable questions (Rajpurkar et al., 2018), longer and more complex passages, Yes/No questions and multi-span answers (Kwiatkowski et al., 2019), and numbers, dates and spans from the question (Dua et al., 2019) as model performance progresses.

Sentiment analysis. The sentiment analysis project is a multi-pronged effort to create a dynamic benchmark for sentiment analysis and to evaluate some of the core hypotheses behind Dynabench. Potts et al. (2021) provide an initial report and the first two rounds of this dataset.

The task is structured as a 3-way classification problem: positive, negative, and neutral. The motivation for using a simple positive/negative dichotomy is to show that there are still very challenging phenomena in this traditional sentiment space. The neutral category was added to avoid (and helped trained models avoid) the false presupposition that every text conveys sentiment information (Pang and Lee, 2008). In future iterations, we plan to consider additional dimensions of sentiment and emotional expression (Alm et al., 2005; Neviarouskaya et al., 2010; Wiebe et al., 2005; Liu et al., 2003; Sudhof et al., 2014).

In this first phase, we examined the question of how best to elicit examples from workers that are diverse, creative, and naturalistic. In the "prompt" condition, we provide workers with an actual sentence from an existing product or service review and ask them to edit it so that it fools the model. In the "no prompt" condition, workers try to write original sentences that fool the model. We find that the "prompt" condition is superior: workers generally make substantial edits, and the resulting sentences are more linguistically diverse than those in the "no prompt" condition.

In a parallel effort, we also collected and validated hard sentiment examples from existing corpora, which will enable another set of comparisons that will help us to refine the Dynabench protocols and interfaces. We plan for the dataset to continue to grow, probably mixing attested examples with those created on Dynabench with the help of prompts. With these diverse rounds, we can address a wide range of question pertaining to dataset artifacts, domain transfer, and overall robustness of sentiment analysis systems.

Hate speech detection. The hate speech task classifies whether a statement expresses hate against a protected characteristic or not. Detecting hate is notoriously difficult given the important role played by context and speaker (Leader Maynard and Benesch, 2016) and the variety of ways in which hate can be expressed (Waseem et al., 2017). Few high-quality, varied and large training datasets are available for training hate detection systems (Vidgen and Derczynski, 2020; Poletto et al., 2020; Vidgen et al., 2019).

We organised four rounds of data collection and model training, with preliminary results reported in (Vidgen et al., 2021). In each round, annotators are tasked with entering content that tricks the model into giving an incorrect classification. The content is created by the annotators and as such is synthetic in nature. At the end of each round the model is retrained and the process is repeated. For the first round, we trained a RoBERTa model on 470,000 hateful and abusive statements[4]. For subsequent rounds the model was trained on the original data plus content from

[4]Derived from https://hatespeechdata.com, in anonymized form.

Task	Rounds	Examples	vMER
NLI	4	170,294	33.24%
QA	2	36,406	33.74%
Sentiment	3	19,975	35.00%
Hate speech	4	41,255	43.90%

Table 4.1: Statistics for the initial four official tasks.

the prior rounds. Due to the complexity of online hate, we hired and trained analysts rather than paying for crowd-sourced annotations. Each analyst was given training, support, and feedback throughout their work.

In all rounds annotators provided a label for whether content is hateful or not. In rounds 2, 3 and 4, they also gave labels for the target (i.e., which group has been attacked) and type of statement (e.g., derogatory remarks, dehumanization, or threatening language). These granular labels help to investigate model errors and improve performance, as well as directing the identification of new data for future entry. For approximately half of entries in rounds 2, 3 and 4, annotators created "perturbations" where the text is minimally adjusted so as to flip the label (Gardner et al., 2020; Kaushik et al., 2020). This helps to identify decision boundaries within the model, and minimizes the risk of overfitting given the small pool of annotators.

Over the four rounds, content becomes increasingly adversarial (shown by the fact that target models have lower performance on later rounds' data) and models improve (shown by the fact that the model error rate declines and the later rounds' models have the highest accuracy on each round). We externally validate performance using the HATECHECK suite of diagnostic tests from (Röttger et al., 2021). We show substantial improvement over the four rounds, and our final round target model achieves 94% on HATECHECK, outperforming the models presented by the original authors.

4.4.3 Dynabenchmarking NLP

Table 4.1 shows an overview of the current situation for the four tasks. Some tasks are further along in their data collection efforts than others. The validated model

error rate (vMER; the number of human-validated model errors divided by the total number of examples—note that the error rates are not necessarily comparable across tasks, since the interfaces and in-the-loop models are not identical) is still very high across all tasks, clearly demonstrating that NLP is far from solved.

4.5 Caveats and Objections

Dynamic Adversarial Data and Distributional Shift. Crowdsourced texts inherently exhibit unnatural qualities due to the artificial nature of the collection setting and the non-representative demographics of crowdworkers. Dynabench, while potentially exacerbating this issue, incorporates features to mitigate it, such as using naturalistic prompts to encourage diverse and realistic data creation. Combining adversarially collected data with non-adversarial or naturally collected data can help capture both average and worst-case scenarios, improving model robustness. Additionally, Dynabench offers a platform to study natural distributional shifts, such as changes in word or phrase meanings over time or across domains.

Annotator Overfitting and Continual Learning. Annotators may inadvertently 'overfit' by focusing on specific model weaknesses, potentially leading to cyclical progress where improved models lose performance on capabilities that were relevant in earlier rounds. To address this, continual learning approaches should focus on understanding distributional shifts and their impact on model learning. Evaluating models across all rounds and high-quality static test sets, potentially with recency-based weighting, ensures comprehensive assessment. Using ensembles of diverse architectures in the loop can further mitigate overfitting to a single model.

Future Models and Benchmark Evolution. While we cannot account for as-yet-undeveloped models, ensembles of current architectures can serve as a reasonable approximation, provided they perform adequately. As observed in Chapter 3, data collected with representative current models can also be extremely valuable for improving the robustness of stronger future models. Benchmark evolution is not unique to Dynabench; datasets like SemEval and WMT have iterated over time. Dynabench's proactive approach to dataset saturation accelerates progress by *antic-*

ipating saturation and continuously updating benchmarks to reflect this.

Generative Tasks and Evaluation. Extending Dynabench to generative tasks requires addressing the lack of ground truth annotations. Discretising generation through multiple-choice formats is one such approach. Relying on annotator judgments to determine model correctness, as explored in Chapter 6, broadens task scope considerably and is seen to be highly effective.

Adversarial Training Data Utility. Chapter 3 and Nie et al. (2020) show that adversarially-collected data is more diverse and also provides performance gains somewhat independent of the model in the loop i.e. there is benefit in involving *any* sufficiently capable model in the annotation loop. However, counterfactually-augmented data does not always enhance generalisation (Huang et al., 2020). Combining adversarial and non-adversarial data during training and testing is recommended. The utility of adversarial data depends on task and model characteristics, necessitating additional research.

Cost Considerations Dynamic benchmarking is more expensive than traditional methods due to the need for model-fooling annotations and validation, and continual updates to the benchmark. However, dynamic datasets may have longer benchmark lifespans, potentially justifying the higher costs. Dynamic Adversarial Data Collection can also be more expensive per example than Standard Data Collection, as seen in Chapter 6, however more research is required to quantify the cost per unit signal rather than the cost per example. Gamification and community engagement initiatives further aim to incentivise contributions.

In summary, Dynabench addresses challenges in data collection and benchmarking through proactive strategies, while acknowledging the need for ongoing research to optimise its utility across diverse tasks and models.

4.6 Conclusion and Outlook

We introduced Dynabench, a research platform for dynamic benchmarking. Dynabench opens up exciting new research directions, such as investigating the effects of ensembles in the loop, distributional shift characterisation, exploring annotator

efficiency, investigating the effects of annotator expertise, and improving model robustness to targeted adversarial attacks in an interactive setting. It also facilitates further study in dynamic data collection, and more general cross-task analyses of human-and-machine interaction. The current iteration of the platform is only just the beginning of a longer journey. In the immediate future, we aim to achieve the following goals:

Anyone can run a task. Having created a tool that allows for human-in-the-loop model evaluation and data collection, we aim to make it possible for anyone to run their own task. To get started, only three things are needed: a target model, a (set of) context(s), and a pool of annotators.

Multilinguality and multimodality. As of now, Dynabench is text-only and focuses on English, but we hope to change that soon.

Live model evaluation. Model evaluation should not be about one single number on some test set. If models are uploaded through a standard interface, they can be scored automatically along many dimensions. We would be able to capture not only accuracy, for example, but also usage of computational resources, inference time, fairness, and many other relevant dimensions. This will in turn enable dynamic leaderboards, for example based on utility (Ethayarajh and Jurafsky, 2020). This would also allow for backward-compatible comparisons, not having to worry about the benchmark changing, and automatically putting new state of the art models in the loop, addressing some of the main objections.

One can easily imagine a future where, in order to fulfill reproducibility requirements, authors do not only link to their open source codebase but also to their model inference point so others can "talk with" their model. This will help drive progress, as it will allow others to examine models' capabilities and identify failures to address with newer even better models. If we cannot always democratize the *training* of state-of-the-art AI models, at the very least we can democratize their *evaluation*.

4.7 Reflection

This chapter introduced Dynabench, an open-source, community-based research platform for dynamic data collection and benchmarking. Dynabench aims to tackle many well-established challenges around model evaluation including saturation, bias, alignment, reproducibility, accessibility, backward compatibility and maximising utility. Along these lines, Dynabench has powered, supported, or otherwise made various research efforts possible including work introducing new datasets or resources, such as:

- *Learning from the Worst: Dynamically Generated Datasets Improve Online Hate Detection* (Vidgen et al., 2021)
- *DynaSent: A Dynamic Benchmark for Sentiment Analysis* (Potts et al., 2021)
- *Human-Adversarial Visual Question Answering* (Sheng et al., 2021)
- *Hatemoji: A Test Suite and Dataset for Benchmarking and Detecting Emoji-based Hate* (Kirk et al., 2022)
- *Adversarial Nibbler: A Data-Centric Challenge for Improving the Safety of Text-to-Image Models* (Parrish et al., 2023)

Furthermore, Dynabench has supported exploration into methodological improvement including:

- *Improving Question Answering Model Robustness with Synthetic Adversarial Data Generation* (Bartolo et al., 2021) – see chapter 5.
- *On the Efficacy of Adversarial Data Collection for Question Answering* (Kaushik et al., 2021)
- *Analyzing Dynamic Adversarial Training Data in the Limit* (Wallace et al., 2022)
- *Models in the Loop: Aiding Crowdworkers with Generative Annotation Assistants* (Bartolo et al., 2022b) – see chapter 6

- *Proceedings of the First Workshop on Dynamic Adversarial Data Collection* (Bartolo et al., 2022a)
- *DataPerf: Benchmarks for Data-Centric AI Development* (Mazumder et al., 2023)

Dynabench has also facilitated investigations and advancements in evaluation approaches, including:

- *To what extent do human explanations of model behavior align with actual behavior?* (Prasad et al., 2021)
- *Dynaboard: A Holistic Evaluation-As-A-Service Benchmarking Platform* (Ma et al., 2021)
- *Findings of the WMT 2021 Shared Task on Large-Scale Multilingual Machine Translation* (Wenzek et al., 2021)
- *Dynatask: A Platform for Creating Dynamic AI Benchmark Tasks* (Thrush et al., 2022)
- *Findings of the BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora* (Warstadt et al., 2023)
- *The PRISM Alignment Project: What Participatory, Representative and Individualised Human Feedback Reveals About the Subjective and Multicultural Alignment of Large Language Models* (Kirk et al., 2024)

Looking forward, Dynabench continues to support the research community through additional features under development including a new architecture for simpler and easier development of creation and validation interfaces, improved documentation, better annotator platform integration, and continued support for new tasks and challenges, with over eight currently in active development.

In the context of Large Language Models, the dynamic nature of Dynabench, whether intentionally or not, continues to drive much of the current narrative. With web-scale data scrapes commonly used for training, the output from similar models

is often included in the training data. This creates a dynamic loop where the models are trained on data that includes their own potential outputs.

From an evaluation perspective, the rapid evolution of test sets is necessary to keep up with the improving performance of these systems and to prevent test example leakage into training data. As a result, we find ourselves in a dynamic operating paradigm, one that Dynabench has deeply modelled and explored.

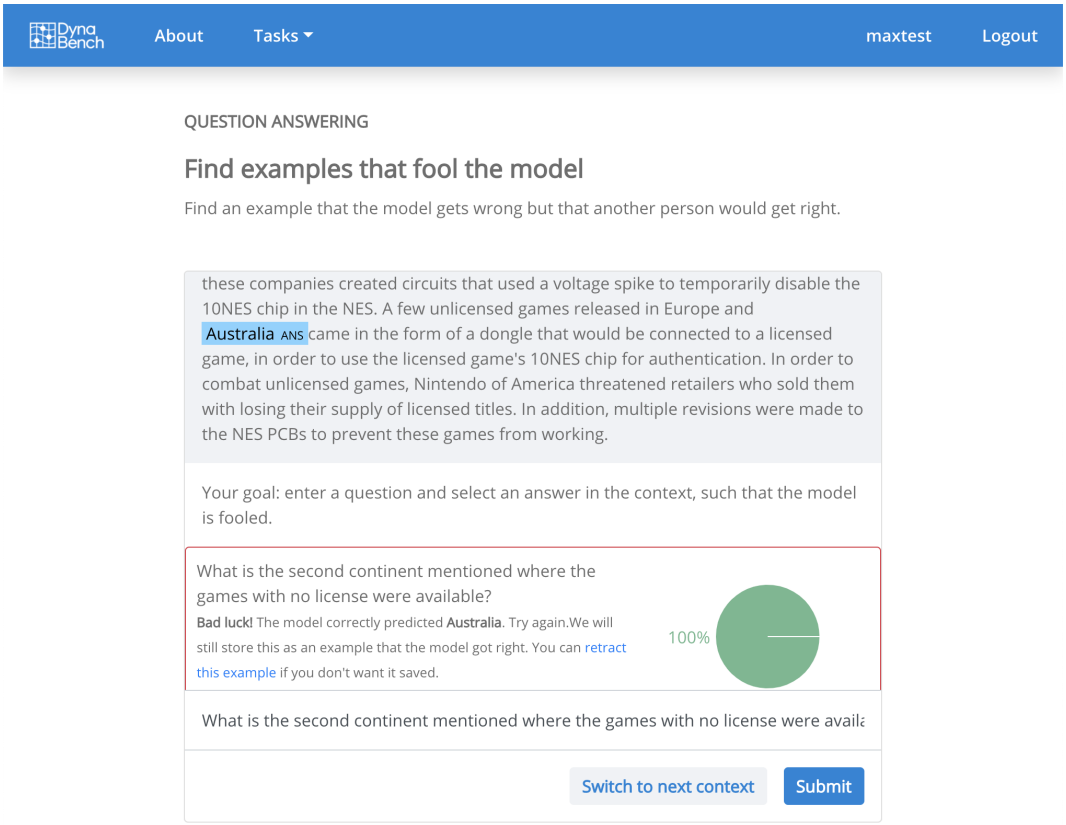


Figure 4.3: The Dynabench annotation interface for the original question answering task, displaying the model's confidence in its predicted answer to the question.

The original Dynabench interfaces also experimented with displaying model confidence scores. An example of this for the original Question Answering task is shown in Figure 4.3, in this case demonstrating a 100% model confidence in the predicted answer "Australia". This feature was designed to assist annotators in their attempts to identify model weaknesses and was effective at doing so on the basis of feedback received — although this was not empirically tested.

We note that since Extractive Question Answering models predict answer can-

didates by classifying start and end indices corresponding to the answer span positions within the passage, a simple estimate of model confidence can be taken using the start and end probabilities assigned by the model. However, the recent popularity of Large Language Models, which do not share this property, makes estimating model confidence more challenging for the most recent systems. We mention the concept of model confidence briefly here since we expand on it as an area of future exploration in Chapter 7.

The Dynabench platform has supported a wide range of research endeavours by offering a broad range of features including user and task management, model upload and hosting, dataset creation and upload, and an Evaluation as a Service (EaaS) approach powered by Dynalab allowing the submission of model code instead of predictions, and evaluation on hosted datasets with constant updates when new datasets are made available. The next two chapters will dive deeper into some of the research contributions that it has facilitated.

Chapter 5

Synthetic Adversarial Data Generation

Dynamic Adversarial Data Collection (DADC), as discussed in the previous two chapters, offers several benefits. It enables the collection of diverse data, improves model performance, enhances adversarial robustness, and facilitates better generalisation. This approach enriches the data collection process and contributes to more robust and adaptable models, but it comes at a cost. DADC is more involved, requires careful annotation setup and incentive structure design, requires deployment of a model-in-the-loop, and is generally more expensive than Standard Data Collection (SDC).

Given these considerations, two intriguing research questions emerged: "What is the optimal mix of standard and adversarially-collected data for training within a fixed budget?", and *"Can we further improve model robustness without additional data collection?"*

Initial experiments provided insights into the first question, suggesting that for best overall performance across evaluation settings, the optimal setting was a roughly balanced half standard and half adversarially-collected data mixture. This finding aligned with expectations, but it was worth noting that the bands of high performance were fairly wide. That is, as a quick takeaway, having a reasonable proportion of standard and adversarial data in the data mixture provided most of the benefit, as shown in Figure 5.1.

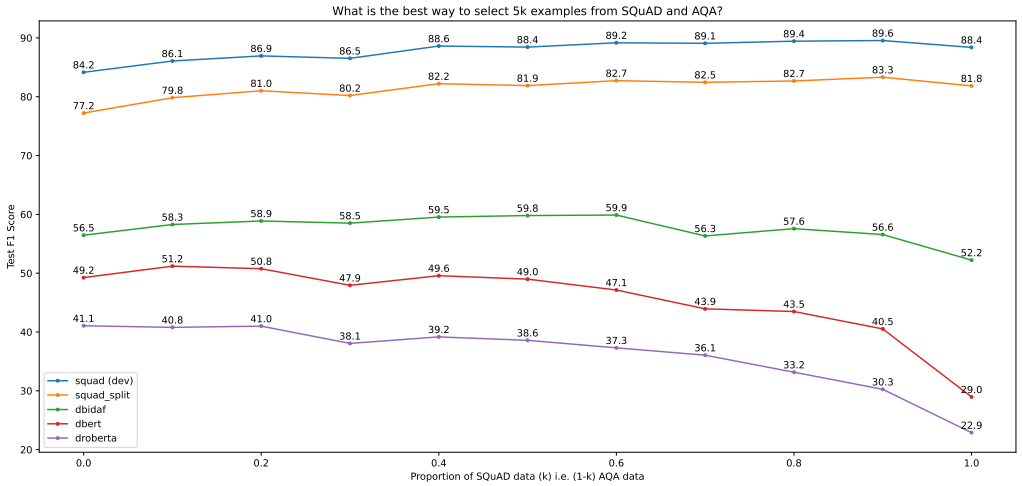


Figure 5.1: Results from experiments exploring the optimal mixture of standard and adversarially-collected data with a fixed 5k example budget. We observe that on the more challenging evaluation sets, the effect of including adversarially-collected data is more pronounced, with the best performance achieved with a roughly balanced mixture of standard and adversarially-collected data. We also note that the performance bands are fairly wide, suggesting that a reasonable proportion of standard and adversarial data in the mixture provides most of the benefit.

Exploring the second research question was more involved and required the development of novel approaches. Existing work in the space of synthetic adversarial data augmentation using generative models was limited. This was especially challenging in the context of extractive Question Answering, where various pipeline components needed to be proposed and improved, such as identifying passages of interest, identifying which spans of text might make good answer candidates, simulated adversarial human behaviour through generated questions, and filtering and rewriting techniques for enhancing synthetic data quality and consistency.

In this chapter, we introduce Synthetic Adversarial Data Generation (SADG) and describe the strategies employed to tackle the challenges mentioned above. By leveraging SADG, we were able to achieve notable enhancements in model performance and considerable robustness improvements, all without incurring any additional human annotation costs.

The material in this chapter is based on the published work titled *"Improving Question Answering Model Robustness with Synthetic Adversarial Data Gen-*

eration" authored by **Max Bartolo**, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp and Douwe Kiela.

This work was published and presented virtually at the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP).

5.1 Overview

Despite recent progress, state-of-the-art question answering models remain vulnerable to a variety of adversarial attacks. While dynamic adversarial data collection, in which a human annotator tries to write examples that fool a model-in-the-loop, can improve model robustness, this process is expensive which limits the scale of the collected data. In this work, we are the first to use synthetic adversarial data generation to make question answering models more robust to human adversaries. We develop a data generation pipeline that selects source passages, identifies candidate answers, generates questions, then finally filters or re-labels them to improve quality. Using this approach, we amplify a smaller human-written adversarial dataset to a much larger set of *synthetic* question-answer pairs. By incorporating our synthetic data, we improve the state-of-the-art on the AdversarialQA dataset by 3.7F_1 and improve model generalisation on nine of the twelve MRQA datasets. We further conduct a novel human-in-the-loop evaluation and show that our models are considerably more robust to new human-written adversarial examples: crowdworkers can fool our model only 8.8% of the time on average, compared to 17.6% for a model trained without synthetic data.

5.2 Introduction

Large-scale labelled datasets like SQuAD (Rajpurkar et al., 2016) and SNLI (Bowman et al., 2015) have been driving forces in natural language processing research. Over the past few years, however, such "statically collected" datasets have been shown to suffer from various problems. In particular, they often exhibit inadvertent spurious statistical patterns that models learn to exploit, leading to poor model robustness and generalisation (Jia and Liang, 2017; Gururangan et al., 2018; Geva et al., 2019; McCoy et al., 2019; Lewis et al., 2021a).

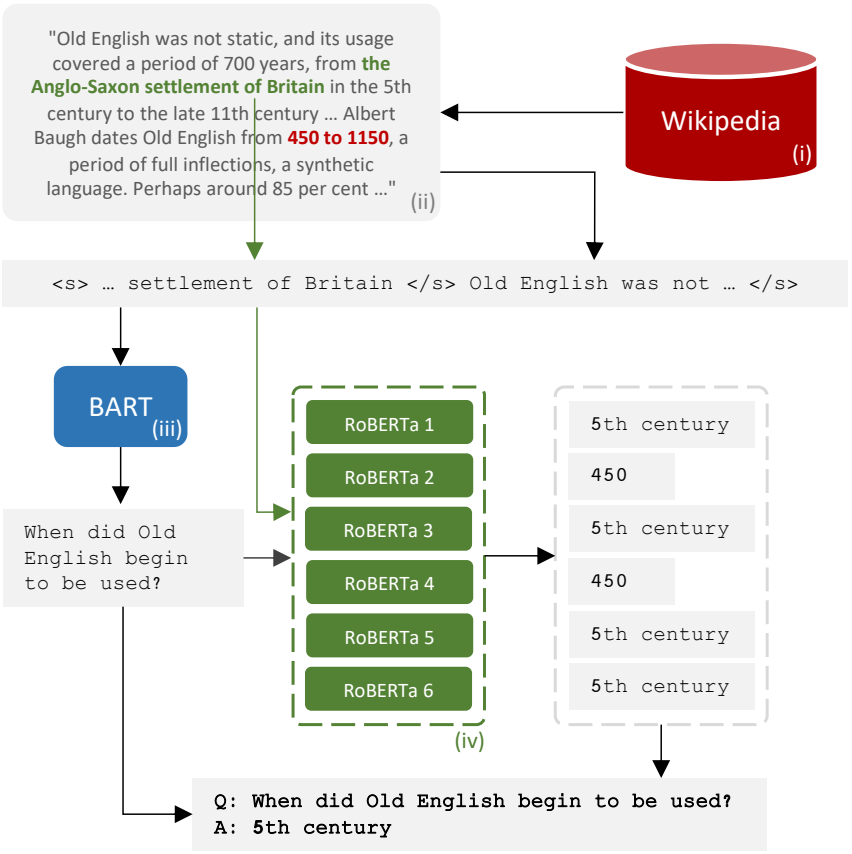


Figure 5.2: The Synthetic Adversarial Data Generation Pipeline showing: (i) passage selection from Wikipedia; (ii) answer candidate selection and filtering by model confidence (an example retained answer shown in green, and a dropped answer candidate in red); (iii) question generation using $\text{BART}_{\text{Large}}$; and (iv) answer re-labelling using self-training. The generated synthetic data is then used as part of the training data for a downstream Reading Comprehension model.

A recently proposed alternative is dynamic data collection (Bartolo et al., 2020; Nie et al., 2020), where data is collected with both humans and models in the annotation loop. Usually, these humans are instructed to ask adversarial questions that fool existing models. Dynamic adversarial data collection is often used to evaluate the capabilities of current state-of-the-art models, but it can also create higher-quality training data (Bartolo et al., 2020; Nie et al., 2020) due to the added incentive for crowdworkers to provide challenging examples. It can also reduce the prevalence of dataset biases and annotator artefacts over time (Bartolo et al., 2020; Nie et al., 2020), since such phenomena can be subverted by model-fooling examples collected in subsequent rounds. However, dynamic data collection can be more

expensive than its static predecessor as creating examples that elicit a certain model response (i.e., fooling the model) requires more annotator effort, resulting in more time spent, and therefore higher cost per example.

In this work, we develop a synthetic adversarial data generation pipeline, making novel contributions to the answer selection, question generation, and filtering and re-labelling tasks. We show that dynamic adversarial data collection can be made more sample efficient by synthetically generating (see Figure 5.2) examples that improve the robustness of models in terms of performance on adversarially-collected datasets, comprehension skills, and domain generalisation.

We are also the first to evaluate models in-the-loop for robustness to human adversaries using the *macro-averaged validated model error rate*, demonstrating considerable improvements with crowdworkers only able to fool the model-in-the-loop 8.8% of the time on average, compared to 17.6% for our best baseline. The collected dataset will form part of the evaluation for a new round of the Dynabench QA task.[1]

5.3 Related Work

5.3.1 Adversarial Data Collection

We directly extend the AdversarialQA dataset introduced in Chapter 3.

While appealing, human-generated adversarial data is expensive to collect; our work is complementary in that it explores methods to extract further value from existing adversarially collected datasets without requiring additional annotation effort.

5.3.2 Synthetic Question Generation

Many approaches have been proposed to generate question-answer pairs given a passage (Du et al., 2017; Du and Cardie, 2018; Zhao et al., 2018; Lewis and Fan, 2019; Alberti et al., 2019; Puri et al., 2020; Lewis et al., 2021b). These generally use a two-stage pipeline that first identifies an answer conditioned on a passage, then generates a question conditioned on the passage and answer; we train a similar pipeline in our work.

[1] https://dynabench.org/tasks/qa

G-DAUG (Yang et al., 2020) trains generative models to synthesise training data for commonsense reasoning. Our work focuses on extractive question-answering (QA), which motivates the need for different generative models. Yang et al. (2020) filter generated examples using influence functions, or methods that attempt to maximise diversity; we find that a different approach that considers answer agreement between QA models trained with different random seeds leads to better performance in our setting.

5.3.3 Self-training

In self-training, a model is trained to both predict correctly on labelled examples and increase its confidence on unlabelled examples. Self-training can yield complementary accuracy gains with pretraining (Du et al., 2020) and can improve robustness to domain shift (Kumar et al., 2020). In our setting, large amounts of unlabelled adversarial-style questions are not readily available, which motivates our use of a question generation model.

5.3.4 Human Evaluation

The ultimate goal of automatic machine learning model evaluation is usually stated as capturing human judgements (Callison-Burch et al., 2006; Hill et al., 2015; Vedantam et al., 2015; Liu et al., 2016). Evaluation with real humans is considered beneficial, but not easily scalable, and as such is rarely conducted in-the-loop. With NLP model capabilities ever improving, adversarial *worst case* evaluation becomes even more pertinent. To our knowledge, this work is the first to compare models explicitly by their adversarial validated model error rate (vMER), which we define in Section 5.5.4.

5.4 Synthetic Data Generation

We develop a synthetic data generation pipeline for QA that involves four stages: passage selection, answer candidate selection, question generation, and synthetic data filtering and re-labelling. Due to the complexity of the system, we study each of these in isolation, and then combine our best identified approaches for the final systems. We evaluate each component both intrinsically and on their contribution

Model	Precision (%)	Recall (%)	F_1 (%)
POS Extended	12.7	65.2	20.7
Noun Chunks	17.4	36.9	22.5
Named Entities	30.3	30.0	27.1
Span Extraction, k=15	22.5	26.6	23.7
$\text{BART}_{\text{ans. only}}$, k=15	27.7	31.3	28.6
SAL (ours)	28.6	44.2	**33.7**

Table 5.1: Answer selection results on aligned test set.

to downstream QA performance on the AdversarialQA test sets and an unseen split of the SQuAD1.1 dev set. The final synthetic data generation pipeline consists of:

1. *Passage selection*: we use passages from Wikipedia for this work.
2. *Answer Candidate selection*: the model identifies spans within the passage that are likely to be answers to a question.
3. *Question Generation*: a generative model is used to generate a question, conditioned on the passage and each answer.
4. *Filtering and Re-labelling*: synthetic question-answer pairs that do not meet the necessary criteria are discarded, or have their answers re-labelled using self-training.

Results for the baseline and overall best performing systems are shown in Table 5.7. Results for $\text{ELECTRA}_{\text{Large}}$ (Clark et al., 2020) showing further performance gains are in Appendix B.10.

5.4.1 Data Generation Pipeline

In order to generate synthetic adversarial examples, we first select passages, then identify candidate answers in those passages, generate corresponding questions for these answers, and then filter or re-label for improved quality based on various criteria.

5.4.1.1 Passage Selection

The text passages we use are sourced from SQuAD (further details can be found in Appendix B.1). We also experiment with using passages external to SQuAD, which

Method	#QA pairs	$\mathcal{D}_{\text{SQuAD}}$		$\mathcal{D}_{\text{BiDAF}}$		$\mathcal{D}_{\text{BERT}}$		$\mathcal{D}_{\text{RoBERTa}}$	
		EM	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1
POS Extended	999,034	53.8	71.4	32.7	46.9	30.8	40.2	20.4	27.9
Noun Chunks	581,512	43.3	63.7	28.7	43.1	22.3	31.4	18.2	27.4
Named Entities	257,857	54.2	69.7	30.5	42.5	26.6	35.4	18.1	24.0
Span Extraction	377,774	64.7	80.1	37.8	53.9	27.7	39.1	16.7	26.9
SAL (ours)	566,730	68.2	**82.6**	43.2	59.3	34.9	45.4	**25.2**	**32.8**
SAL threshold (ours)	393,164	**68.5**	82.0	**46.0**	**60.3**	**36.5**	**46.8**	24.2	32.4

Table 5.2: Downstream test results for a $\text{RoBERTa}_{\text{Large}}$ QA model trained on synthetic data generated using different answer selection methods combined with a $\text{BART}_{\text{Large}}$ question generator (trained on SQuAD_{10k} + $\mathcal{D}_{\text{AQA}}$).

are also sourced from Wikipedia. To preserve evaluation integrity, we analyse the 8-gram overlap of all external passages to the evaluation datasets, after normalisation to lower-cased alphanumeric words with a single space delimiter (Radford et al., 2019). We find that just 0.3% of the external passages have any overlap with the evaluation sets, and filter these out.

5.4.1.2 Answer Candidate Selection

The next step is to identify which spans of text within the passages are likely to be answers to a question. We investigate a range of existing methods for answer candidate selection, which takes the passage as input and outputs a set of possible answers. We further propose a self-attention-based classification head that jointly models span starts and ends, with improved performance.

Since SQuAD and the AdversarialQA datasets use the same passages partitioned into the same data splits, we align the annotated answers to create representative answer selection training, validation and test sets. Dataset statistics (see Appendix B.3), highlight the high percentage of overlapping answers suggesting that existing answer tagging methods (Zhou et al., 2017; Zhao et al., 2018) might struggle, and models should ideally be capable of handling span overlap.

Baseline Systems We investigate three baseline systems; noun phrases and named entities following Lewis et al. (2019), as well as an extended part-of-speech tagger incorporating named entities, adjectives, noun phrases, numbers, distinct proper nouns, and clauses.

Span Extraction We fine-tune a $\text{RoBERTa}_{\text{Large}}$ span extraction model as investigated in previous work (Alberti et al., 2019; Lewis and Fan, 2019). We treat the number of candidates to sample as a hyper-parameter and select the optimal value for $k \in \{1, 5, 10, 15, 20\}$ on the validation set.

Generative Answer Detection We use $\text{BART}_{\text{Large}}$ (Lewis et al., 2020) in two settings; one generating answer and question, and the other where we generate the answer only, as we find that this setting provides better control of answer diversity. We use the same range of $k \in \{1, 5, 10, 15, 20\}$ for both settings.

Self-Attention Labelling (SAL) We propose a multi-label classification head to jointly model candidate start and end tokens, and provide a binary label for whether each possible span of text from the passage is a candidate answer. We adapt scaled dot-product attention (Vaswani et al., 2017) where the candidate start, **S**, and end, **E**, token representations are analogous to the projected layer input queries and keys. We apply a sigmoid over the computed attention scores, giving a matrix where each cell gives the probability $p(a_{ij}|c)$ of whether the span in the context, c, with start index i and end index j is a valid answer candidate. Formally:

$$p(a_{ij}|c) = \sigma\left(\frac{\sum_{k=1}^{d} s_{ik} e_{kj}}{\sqrt{d}}\right)$$

We optimise using binary cross-entropy, masking out impossible answer spans defined as those not in the passage, with end indices before start, or longer than the maximum permitted answer length, and upweigh positive examples to help counteract the class imbalance. We decode from the output probability matrix to the original passage tokens using a reversible tokeniser and use a probability threshold of 0.5 for candidate selection, which can be adapted to tune precision and recall.

While answer candidate selection only requires a single attention head, the multi-head implementation allows application to any labelling task requiring span modelling with overlaps, where each head is trained to predict labels for each class, such as for nested Named Entity Recognition. We implement this in *Transformers* (Wolf et al., 2020) and fine-tune $\text{RoBERTa}_{\text{Large}}$ with SAL on the answer selection dataset.

Evaluation We evaluate performance on the answer selection dataset using entity-level precision, recall, and F_1 on unique normalised candidates. Results are shown in Table 5.1. We further investigate the effects of different answer candidate selection methods on downstream QA model performance (see Table 5.2) by training a $\text{RoBERTa}_{\text{Large}}$ model on synthetic QA pairs generated when using different answer selection methods. To eliminate generated dataset size as a potential confounder, we also replicate these experiments using a sample of 87,000 examples and find similar results (see Appendix B.3).

5.4.1.3 Question Generation

Once answer candidates have been identified for a selected passage, we then generate a corresponding question by directly fine-tuning a $\text{BART}_{\text{Large}}$ (Lewis et al., 2020) autoregressive sequence generation decoder.[2] To discourage the model from memorising the questions in the SQuAD training set and directly reproducing these, we train on a subset of 10k examples from SQuAD, selected such that they correspond to the same source passages as the AdversarialQA training data. This ensures that when scaling up synthetic generation, the vast majority of passages are previously completely unseen to the generator.

Source Questions Since the types of questions a generative model is trained on can impact both performance and diversity, we experiment with training on SQuAD and different subsets of AdversarialQA, and the combination of both. Examples of the generated questions are shown in Table 5.3.

We carry out a manual answerability analysis on a random sample of 30 generated questions (using beam search with $k = 5$) in each of these settings (see Table 5.4 and Appendix B.2). We define answerability by the following criteria: (i) The question must be answerable from a single continuous span in the passage; (ii) There must be only one valid (or clearly one most valid) answer (e.g. in the case of a co-reference the canonical entity name should be the answer); (iii) A human should be able to answer the question correctly given sufficient time; and (iv) The

[2] We also try generating multiple questions but consistently find that generating one question per answer provides the best downstream results despite the additional data.

Context: *Following the series revival in 2005, Derek Jacobi $_{ANS}$ provided the character's re-introduction in the 2007 episode "Utopia". During that story the role was then assumed by John Simm who returned to the role multiple times through the Tenth Doctor's tenure. As of the 2014 episode "Dark Water," it was revealed that the Master had become a female incarnation or "Time Lady," going by the name of "Missy", played by Michelle Gomez.*

SQuAD_{10k}	Who portrayed the Master in the 2007 episode "Utopia"?
$\mathcal{D}_{\text{BiDAF}}$	Who replaced John Simm as the Tenth Doctor? **(Answer Mismatch)**
$\mathcal{D}_{\text{BERT}}$	Who played the Master in the 2007 episode "Utopia"?
$\mathcal{D}_{\text{RoBERTa}}$	Who was the first actor to play the Master?
$\mathcal{D}_{\text{AQA}}$	Who played the Master first, Derek Jacobi or John Simm?
$\text{SQuAD}_{10k} + \mathcal{D}_{\text{AQA}}$	Who re-introduced the character of the Master?

Table 5.3: Examples of questions generated using BART trained on different source datasets.

Model	**Valid**	**Answer Mismatch**	**Ungrammatical**	**Invalid**
SQuAD_{10k}	90.0%	10.0%	0.0%	0.0%
$\mathcal{D}_{\text{BiDAF}}$	70.0%	30.0%	0.0%	0.0%
$\mathcal{D}_{\text{BERT}}$	76.7%	23.3%	0.0%	0.0%
$\mathcal{D}_{\text{RoBERTa}}$	70.0%	20.0%	0.0%	10.0%
$\mathcal{D}_{\text{AQA}}$	76.7%	16.7%	0.0%	6.7%
SQuAD_{10k}+$\mathcal{D}_{\text{AQA}}$	93.3%	6.7%	0.0%	0.0%

Table 5.4: Manual analysis of questions generated when training on different source data.

correct answer is the one on which the model was conditioned during question generation. We find that when the models attempt to generate complex questions, the generated question is often inconsistent with the target answer, despite remaining well-formed. We also observe that when the generated question requires external knowledge (e.g. "What is a tribe?" or "Which is not a country?") the models are reasonably consistent with the answer, however, they often lose answer consistency when answering the question requires resolving information in the passage (e.g. "What is the first place mentioned?").

For each of these models, we generate 87k examples (the same size as the SQuAD training set to facilitate comparison) using the human-provided answers, and then measure the effects on downstream performance by training a QA model

Method	#QA pairs	$\mathcal{D}_{\text{SQuAD}}$		$\mathcal{D}_{\text{BiDAF}}$		$\mathcal{D}_{\text{BERT}}$		$\mathcal{D}_{\text{RoBERTa}}$	
		EM	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1
R_{SQuAD}	87,599	73.2	86.3	48.9	64.3	31.3	43.5	16.1	26.7
$\text{R}_{\text{SQuAD+AQA}}$	117,599	74.2	86.9	57.4	72.2	53.9	65.3	43.4	54.2
SQuAD_{10k}	87,598	69.2	82.6	37.1	52.1	22.4	32.3	13.9	22.3
$\mathcal{D}_{\text{BiDAF}}$	87,598	67.1	80.4	41.4	56.5	**33.1**	43.8	22.0	32.5
$\mathcal{D}_{\text{BERT}}$	87,598	67.4	80.2	36.3	51.1	30.3	40.6	18.8	29.5
$\mathcal{D}_{\text{RoBERTa}}$	87,598	63.4	77.9	32.6	47.9	27.2	37.5	20.6	32.0
$\mathcal{D}_{\text{AQA}}$	87,598	65.5	80.1	37.0	53.0	31.1	40.9	**23.2**	**33.3**
SQuAD_{10k} + $\mathcal{D}_{\text{AQA}}$	87,598	**71.9**	**84.7**	**44.1**	**58.8**	32.9	**44.1**	19.1	28.8

Table 5.5: Downstream QA test results using generative models trained on different source data. We compare these results to baseline RoBERTa models trained on SQuAD, and on the combination of SQuAD and AdversarialQA.

on this synthetic data. Results are shown in Table 5.5. We find that, in this setting, the best source data for the generative model is consistently the combination of SQuAD and AdversarialQA. We also note that using only synthetic generated data, we can achieve good performance on $\mathcal{D}_{\text{SQuAD}}$ consistent with the findings of Puri et al. (2020), and outperform the model trained on the human-written SQuAD data on $\mathcal{D}_{\text{BERT}}$ (+0.6F_1) and $\mathcal{D}_{\text{RoBERTa}}$ (+6.6F_1). This is in line with the observations in Chapter 3 suggesting that the distribution of the questions collected using progressively stronger models-in-the-loop is less similar to that of SQuAD. It also shows that the generator can successfully identify and reproduce patterns of adversarially-written questions. However, the results using synthetic data alone are considerably worse than when training the QA model on human-written adversarial data with, for example, a performance drop of 21.2F_1 for $\mathcal{D}_{\text{BERT}}$. This suggests that while we can do well on SQuAD using synthetic questions alone, we may need to combine the synthetic data with the human-written data for best performance in the more challenging adversarial settings.

Question Diversity In order to provide training signal diversity to the downstream QA model, we experiment with a range of decoding techniques (see Appendix B.4), and then evaluate these by downstream performance of a QA model trained on the questions generated in each setting. We observe minimal variation in downstream performance as a result of question decoding strategy, with the best downstream

Filtering Method	#QA pairs	$\mathcal{D}_{\text{SQuAD}}$		$\mathcal{D}_{\text{BiDAF}}$		$\mathcal{D}_{\text{BERT}}$		$\mathcal{D}_{\text{RoBERTa}}$	
		EM	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1
Answer Candidate Conf. ($thresh = 0.6$)	362,281	68.4	82.4	42.9	57.9	36.3	45.9	28.0	36.5
Question Generator Conf. ($thresh = 0.3$)	566,725	69.3	83.1	43.5	58.9	36.3	46.6	26.2	34.8
Influence Functions	288,636	68.1	81.9	43.7	58.6	36.1	46.6	27.4	36.4
Ensemble Roundtrip Consistency (6/6 correct)	250,188	74.2	86.2	55.1	67.7	45.8	54.6	31.9	40.3
Self-training (ST)	528,694	74.8	87.0	53.9	67.9	47.5	57.6	35.2	44.6
Answer Candidate Conf. ($thresh = 0.5$) & ST	380,785	**75.1**	**87.0**	**56.5**	**70.0**	**47.9**	**58.7**	**36.0**	**45.9**

Table 5.6: Downstream QA test results for different filtering strategies, showing best hyper-parameter settings.

results obtained using nucleus sampling ($top_p = 0.75$). However, we also obtain similar downstream results with standard beam search using a beam size of 5. We find that, given the same computational resources, standard beam search is roughly twice as efficient, and therefore opt for this approach for our following experiments.

5.4.1.4 Filtering and Re-labelling

The synthetic question generation process can introduce various sources of noise, as seen in the previous analysis, which could negatively impact downstream results. To mitigate these effects, we explore a range of filtering and re-labelling methods. Results for the best performing hyper-parameters of each method are shown in Table 5.6 and results controlling for dataset size are in Appendix B.5.

Answer Candidate Confidence. We select candidate answers using SAL (see section 5.4.1.2), and filter based on the span extraction confidence of the answer candidate selection model, estimated as the joint start and end token probabilities.

Question Generator Confidence. We filter out samples below various thresholds of the probability score assigned to the generated question sequence, taking the product across tokens, by the question generation model.

Influence Functions. We use influence functions (Cook and Weisberg, 1982; Koh and Liang, 2017) to estimate the effect on the validation loss of including a synthetic example as explored by Yang et al. (2020), but adapted for QA. We filter out examples estimated to increase the validation loss.

Ensemble Roundtrip Consistency. Roundtrip consistency (Alberti et al., 2019; Fang et al., 2020) uses an existing fine-tuned QA model to attempt to answer the

generated questions, ensuring that the predicted answer is consistent with the target answer prompted to the generator. Since our setup is designed to generate questions which are intentionally challenging for the QA model to answer, we attempt to exploit the observed variation in model behaviour over multiple random seeds, and replace the single QA model with a six-model ensemble. We find that filtering based on the number of downstream models that correctly predict the original target answer for the generated question produces substantially better results than relying on the model confidence scores, which could be prone to calibration imbalances across models.

Self-training. Filtering out examples that are not roundtrip-consistent can help eliminate noisy data, however, it also results in (potentially difficult to answer) questions to which a valid answer may still exist being unnecessarily discarded. Self-training has been shown to improve robustness to domain shift (Kumar et al., 2020) and, in our case, we re-label answers to the generated questions based on the six QA model predictions.

Specifically, in our best-performing setting, we keep any examples where at least five of the six QA models agree with the target answer (i.e. the one with which the question generator was originally prompted), re-label the answers for any examples where at least two of the models QA agree among themselves, and discard the remaining examples (i.e. those for which there is no agreement between any of the QA models).

We find that the best method combines self-training with answer candidate confidence filtering. By using appropriate filtering of the synthetic generated data, combined with the ability to scale to many more generated examples, we approach the performance of $R_{\text{SQuAD+AQA}}$, practically matching performance on SQuAD and reducing the performance disparity to just 2.2F_1 on $\mathcal{D}_{\text{BiDAF}}$, 6.6$F_1$ on $\mathcal{D}_{\text{BERT}}$, and 8.3$F_1$ on $\mathcal{D}_{\text{RoBERTa}}$, while still training solely on synthetic data.

5.4.2 End-to-end Synthetic Data Generation

We also try using BART to both select answers and generate questions in an end-to-end setting. We experiment with different source datasets, number of genera-

Model	Training Data	$\mathcal{D}_{\text{BiDAF}}$		$\mathcal{D}_{\text{BERT}}$		$\mathcal{D}_{\text{RoBERTa}}$		**mvMER***
		EM	F_1	*EM*	F_1	*EM*	F_1	%
R_{SQuAD}	SQuAD	$48.6_{1.3}$	$64.2_{1.5}$	$30.9_{1.3}$	$43.3_{1.7}$	$15.8_{0.9}$	$26.4_{1.3}$	20.7%
$R_{\text{SQuAD+AQA}}$	↑ + AQA	$59.6_{0.5}$	$73.9_{0.5}$	$54.8_{0.7}$	$64.8_{0.9}$	$41.7_{0.6}$	$53.1_{0.8}$	17.6%
SynQA	↑ + $\text{SynQA}_{\text{SQuAD}}$	$62.5_{0.9}$	$76.0_{1.0}$	$58.7_{1.4}$	$68.3_{1.4}$	$46.7_{1.8}$	$\mathbf{58.0}_{1.8}$	**8.8%**
$\text{SynQA}_{\text{Ext}}$	↑ + $\text{SynQA}_{\text{Ext}}$	$\mathbf{62.7}_{0.6}$	$\mathbf{76.2}_{0.5}$	$\mathbf{59.0}_{0.7}$	$\mathbf{68.9}_{0.5}$	$\mathbf{46.8}_{0.5}$	$57.8_{0.8}$	12.3%

Table 5.7: Test set results for $\text{RoBERTa}_{\text{Large}}$ trained on different datasets, and augmented with synthetic data. AQA is the AdversarialQA data consisting of the combined $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, and $\mathcal{D}_{\text{RoBERTa}}$ from Chapter 3. We report the mean and standard deviation (subscript) over 6 runs with different random seeds. mvMER is the macro-averaged validated model error rate in the adversarial human evaluation setting (*lower is better).

tions per passage, and decoding hyper-parameters, but our best results fall short of the best pipeline approach at 62.7/77.9 EM/F_1 on $\mathcal{D}_{\text{SQuAD}}$, 30.8/47.4 on $\mathcal{D}_{\text{BiDAF}}$, 23.6/35.6 on $\mathcal{D}_{\text{BERT}}$, and 18.0/28.3 on $\mathcal{D}_{\text{RoBERTa}}$. These results are competitive when compared to some of the other answer candidate selection methods we explored, however, fall short of the results obtained when using SAL. We find that this approach tends to produce synthetic examples with similar answers, but leave exploring decoding diversity to future work.

5.4.3 Fine-tuning Setup

We investigate two primary fine-tuning approaches: combining all training data, and a two-stage set-up in which we first fine-tune on the generated synthetic data, and then perform a second-stage of fine-tuning on the SQuAD and AdversarialQA human-written datasets. Similar to Yang et al. (2020), we find that two-stage training marginally improves performance over standard mixed training, and we use this approach for all subsequent experiments.

5.5 Measuring Model Robustness

Based on the findings in the previous section, we select four final models for robustness evaluation:

1. R_{SQuAD}: using the SQuAD1.1 training data.

2. $R_{\text{SQuAD+AQA}}$: trained on SQuAD combined and shuffled with AdversarialQA.

3. SynQA: uses a two-stage fine-tuning approach, first trained on 314,811 synthetically generated questions on the passages in the SQuAD training set, and then further fine-tuned on SQuAD and AdversarialQA.

4. SynQA$_{\text{Ext}}$: first trained on the same synthetic SQuAD examples as (iii) combined with 1.5M synthetic questions generated on the previously described Wikipedia passages external to SQuAD, and then further fine-tuned on SQuAD and AdversarialQA.

Individual models are selected for the best combined and equally-weighted performance on a split of the SQuAD validation set and all three AdversarialQA validation sets.

We first evaluate model robustness using three existing paradigms: adversarially-collected datasets, checklists, and domain generalisation. We also introduce adversarial human evaluation, a new way of measuring robustness with direct interaction between the human and model.

5.5.1 Adversarially-collected Data

We evaluate the final models on AdversarialQA, with results shown in Table 5.7. We find that synthetic data augmentation yields state-of-the-art results on AdversarialQA, providing performance gains of 2.3F$_1$ on $\mathcal{D}_{\text{BiDAF}}$, 4.1F$_1$ on $\mathcal{D}_{\text{BERT}}$, and 4.9F$_1$ on $\mathcal{D}_{\text{RoBERTa}}$ over the baselines while retaining good performance on SQuAD, a considerable improvement at no additional annotation cost.

5.5.2 Comprehension Skills

CheckList (Ribeiro et al., 2020) is a model agnostic approach that serves as a convenient test-bed for evaluating what *comprehension skills* a QA model could learn. We find that some skills that models struggle to learn when trained on SQuAD, such as discerning between profession and nationality, or handling negation in questions, can be learnt by incorporating adversarially-collected data during training (see Appendix B.8). Furthermore, augmenting with synthetic data improves performance on a variety of these skills, with a 1.7% overall gain for SynQA and 3.1% for SynQA$_{\text{Ext}}$. Adding the external synthetic data improves performance on most

MRQA in-domain

Model	SQuAD		NewsQA		TriviaQA		SearchQA		HotpotQA		NQ		Avg	
	EM	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1
R_{SQuAD}	$84.1_{1.3}$	$90.4_{1.3}$	$41.0_{1.2}$	$57.5_{1.6}$	$60.2_{0.7}$	$69.0_{0.8}$	$16.0_{1.8}$	$20.8_{2.7}$	$53.6_{0.8}$	$68.9_{0.8}$	$40.5_{2.7}$	$58.5_{2.0}$	49.2	60.9
$R_{SQuAD+AQA}$	$84.4_{1.0}$	$90.2_{1.1}$	$41.7_{1.6}$	$58.0_{1.7}$	$\mathbf{62.7}_{0.4}$	$\mathbf{70.8}_{0.3}$	$20.6_{2.9}$	$25.5_{3.6}$	$56.3_{1.1}$	$72.0_{1.0}$	$54.4_{0.5}$	$68.7_{0.4}$	53.3	64.2
SynQA	$88.8_{0.3}$	$\mathbf{94.3}_{0.2}$	$42.9_{1.6}$	$60.0_{1.4}$	$62.3_{1.1}$	$70.2_{1.1}$	$23.7_{3.7}$	$29.5_{4.4}$	$\mathbf{59.8}_{1.1}$	$75.3_{1.0}$	$55.1_{1.0}$	$68.7_{0.8}$	55.4	66.3
$SynQA_{Ext}$	$\mathbf{89.0}_{0.3}$	$\mathbf{94.3}_{0.2}$	$\mathbf{46.2}_{0.9}$	$\mathbf{63.1}_{0.8}$	$58.1_{1.8}$	$65.5_{1.9}$	$\mathbf{28.7}_{3.2}$	$\mathbf{34.3}_{4.1}$	$59.6_{0.6}$	$\mathbf{75.5}_{0.4}$	$\mathbf{55.3}_{1.1}$	$\mathbf{68.8}_{0.9}$	**56.2**	**66.9**

MRQA out-of-domain

Model	BioASQ		DROP		DuoRC		RACE		RelationExt.		TextbookQA		Avg	
	EM	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1
R_{SQuAD}	$53.2_{1.1}$	$68.6_{1.4}$	$39.8_{2.6}$	$52.7_{2.2}$	$49.3_{0.7}$	$60.3_{0.8}$	$35.1_{1.0}$	$47.8_{1.2}$	$74.1_{3.0}$	$84.4_{2.9}$	$35.0_{3.8}$	$44.2_{3.7}$	47.7	59.7
$R_{SQuAD+AQA}$	$54.6_{1.2}$	$\mathbf{69.4}_{0.8}$	$59.8_{1.3}$	$68.4_{1.5}$	$\mathbf{51.8}_{1.1}$	$\mathbf{62.2}_{1.0}$	$38.4_{0.9}$	$51.6_{0.9}$	$75.4_{2.3}$	$85.8_{2.4}$	$40.1_{3.1}$	$48.2_{3.6}$	53.3	64.3
SynQA	$\mathbf{55.1}_{1.5}$	$68.7_{1.2}$	$64.3_{1.5}$	$72.5_{1.7}$	$51.7_{1.3}$	$62.1_{0.9}$	$\mathbf{40.2}_{1.2}$	$\mathbf{54.2}_{1.3}$	$78.1_{0.2}$	$87.8_{0.2}$	$40.2_{1.3}$	$49.2_{1.5}$	**54.9**	**65.8**
$SynQA_{Ext}$	$54.9_{1.3}$	$68.5_{0.9}$	$\mathbf{64.9}_{1.1}$	$\mathbf{73.0}_{0.9}$	$48.8_{1.2}$	$58.0_{1.2}$	$38.6_{0.4}$	$52.2_{0.6}$	$\mathbf{78.9}_{0.4}$	$\mathbf{88.6}_{0.2}$	$\mathbf{41.4}_{1.1}$	$\mathbf{50.2}_{1.0}$	54.6	65.1

Table 5.8: Domain generalisation results on the in-domain (top) and out-of-domain (bottom) subsets of MRQA.

taxonomy-related skills, considerably so on "profession vs nationality", as well as skills such as "his/her" coreference, or subject/object distinction. While many of these skills seem to be learnable, it is worth noting the high variation in model performance over multiple random initialisations.

5.5.3 Domain Generalisation

We evaluate domain generalisation of our final models on the MRQA (Fisch et al., 2019) dev sets, with results shown in Table 5.8.[3] We find that augmenting training with synthetic data provides performance gains on nine of the twelve tasks. Performance improvements on some of the tasks can be quite considerable (up to 8.8F_1 on SearchQA), which does not come at a significant cost on the three tasks where synthetic data is not beneficial.

5.5.4 Adversarial Human Evaluation

While existing robustness measures provide valuable insight into model behaviour, they fail to capture how robust a model might be in a production setting. We use Dynabench (Kiela et al., 2021), a research platform for dynamic benchmarking and evaluation, to measure model robustness in an adversarial human evaluation set-

[3] We note that our results are not directly comparable to systems submitted to the MRQA shared task, which were trained on six "in-domain" datasets; we simply reuse the MRQA datasets for evaluation purposes.

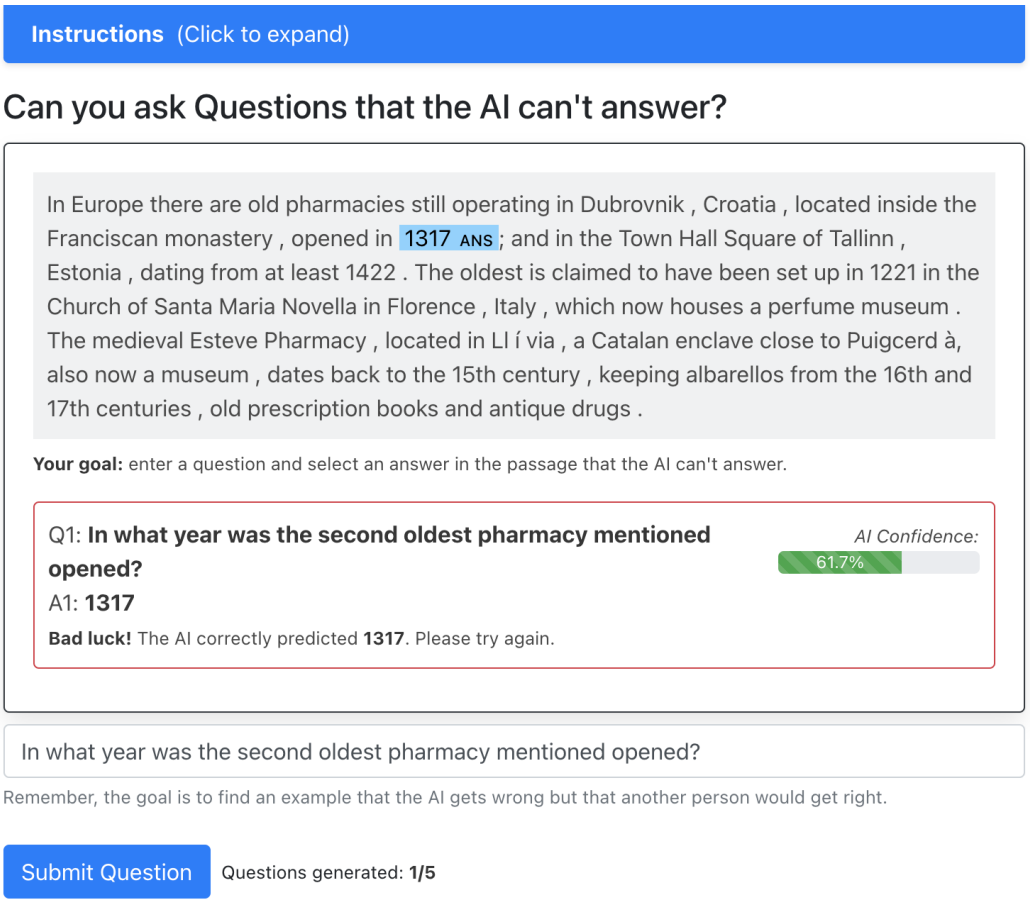


Figure 5.3: The Adversarial Human Evaluation Interface.

ting. This allows for live interaction between the model and human annotator, and more closely simulates realistic and challenging scenarios a deployed system might encounter, compared to evaluation on static datasets.

We set up the experiment as a randomised controlled trial where annotators are randomly allocated to interact with each of our four final models based on a hash of their annotator identifier. We run the experiment through Amazon Mechanical Turk (AMT) using Mephisto.[4] Workers (see Appendix B.9) are first required to complete an onboarding phase to ensure familiarity with the interface, and are then required to ask five questions of the model. We pay $0.20 per question and given a strong incentive to try to beat the model with a $0.50 bonus for each validated question that the model fails to answer correctly.[5] The model identity is kept hidden and

[4] github.com/facebookresearch/Mephisto

[5] Our evaluation setup is different to "Beat the AI" where annotators couldn't submit unless they

workers are awarded an equal base pay irrespective of the model-in-the-loop to avoid creating an incentive imbalance. Each annotator is allowed to write at most 50 questions, to avoid having a few productive annotators dominate our findings. All model-fooling examples are further validated by an expert annotator. We skip validation of questions the model answered correctly, as manual validation of a sample of 50 such examples found that all are valid, suggesting that the QA model's ability to answer them is a good indicator of their validity.

We measure performance as the validated model error rate (vMER), that is, the percentage of validated examples that the model fails to answer correctly. Despite limiting the number of collected examples to 50 per annotator, there is still the potential of an imbalance in the number of QA pairs produced by each annotator. In order to eliminate annotator effect as a potential confounder, we propose using the macro-averaged validated model error rate (mvMER) over annotators, defined as:

$$\text{mvMER} = \frac{1}{n_{ann}} \sum_{i=1}^{n_{ann}} \frac{\text{validated model errors}_i}{\text{number of examples}_i}$$

We find that SynQA roughly halves the model error rate compared to $\text{R}_{\text{SQuAD+AQA}}$ from 17.6% to 8.8% (see Table 5.7, further details in Appendix B.9), meaning that it is considerably harder for human adversaries to ask questions that the model cannot answer. While $\text{SynQA}_{\text{Ext}}$ still considerably outperforms $\text{R}_{\text{SQuAD+AQA}}$ at a 12.3% mvMER, we find that it is not as hard to beat as SynQA in this setting. A low model error rate also translates into increased challenges for the adversarial human annotation paradigm as the effort required for each model-fooling example increases, and provides motivation to expand the current extractive QA task beyond single answer spans on short passages.

These findings further suggest that while static adversarial benchmarks are a good evaluation proxy, performance gains on these may be underestimating the effect on model robustness in a setting involving direct interaction between the models-in-the-loop and human adversaries.

---

beat the model a certain number of times. This creates a different annotation dynamic that we believe is better suited for model evaluation.

5.6 Discussion and Conclusion

In this work, we develop a synthetic adversarial data generation pipeline for QA, identify the best components, and evaluate on a variety of robustness measures. We propose novel approaches for answer candidate selection, adversarial question generation, and synthetic example filtering and re-labelling, demonstrating improvements over existing methods. Furthermore, we evaluate the final models on three existing robustness measures and achieve state-of-the-art results on AdversarialQA, improved learnability of various comprehension skills for CheckList, and improved domain generalisation for the suite of MRQA tasks.

We then put the synthetically-augmented models back in-the-loop in an adversarial human evaluation setting to assess whether these models are actually harder for a human adversary to beat.

We find that our best synthetically-augmented model is roughly twice as hard to beat. Our findings suggest that synthetic adversarial data generation can be used to improve QA model robustness, both when measured using standard methods and when evaluated directly against human adversaries.

Looking forward, the methods explored in this work could also be used to scale the dynamic adversarial annotation process in multiple ways. Synthetic adversarial data generation could facilitate faster iteration over rounds of adversarial human annotation as it reduces the amount of human data required to effectively train an improved QA model. Generative models could also help guide or inspire human annotators as they try to come up with more challenging examples. Furthermore, while our work focuses on improving adversarial robustness, this approach is not limited to the adversarial setting. We believe that our findings can motivate similar investigations for tasks where data acquisition can be challenging due to limited resources, or for improving different aspects of robustness, for example for model bias mitigation.

5.7 Reflection

The work presented in this chapter produced various resources made available to the community. These include:

- The *SynQA* Dataset: Consisting of 314,811 synthetically generated questions on the passages in SQuAD1.1 training set and corresponding answers, this dataset was made publicly available under an MIT license on the Hugging Face datasets hub at https://huggingface.co/datasets/mbartolo/synQA.

- Question Generation Models: Our BART-Large based question generation models were developed in Facebook Research's *fairseq* library and made available at https://github.com/maxbartolo/synQA-question-generators and have been used to generate data for various other research experiments.

- Question Answering Models: Three new state-of-the-art (at the time of release) question answering models were also released on the Hugging Face hub and are available at https://huggingface.co/mbartolo/roberta-large-synqa, https://huggingface.co/mbartolo/roberta-large-synqa-ext (most popular), and https://huggingface.co/mbartolo/electra-large-synqa. Combined, they have been downloaded over 18,000 times and have contributed to various research and real-world use-cases, including powering additional rounds of the Dynabench QA task as well as the DADC Workshop Shared Task and the work that will be described in the next chapter.

- To the best of our knowledge, the answer candidate selection approach remains one of the best-performing methods to date, and is being further explored in at least one ongoing research effort.

Along with improved adversarial robustness, enhanced comprehension skills and better domain generalisation achieved by training on data generated through

SADG, we find that synthetic data collected with a specific model-in-the-loop benefits not only that model but also transfers positively to stronger models. We see evidence of this in Table B.8, where ELECTRA$_{\text{Large}}$, a generally stronger model than RoBERTa$_{\text{Large}}$, trained on SQuAD and AdversarialQA with no synthetic augmentation, performs similarly to the best synthetically augmented RoBERTa$_{\text{Large}}$ models. When we introduce the synthetic adversarial data, we observe similar performance gains as those seen for RoBERTa$_{\text{Large}}$, particularly on the more challenging questions. This finding suggests that the advantages of SADG extend beyond the specific model and setup used for data collection and are, in a sense, future-proof – where data created using a contemporary model can still be used to enhance the performance of future, more davanced models. This finding aligns with the observations from Chapter 3, where adversarial data collected with a weaker model in the loop, such as BiDAF, provided various performance, robustness and generalisation benefits when used to train stronger models (such as BERT or RoBERTa). This transferability underscores the effectiveness of SADG, making it a valuable technique for both current and future model development.

The concepts and techniques presented in this work have also influenced a range of efforts to enhance the robustness of Large Language Models (LLMs). Specifically, the ideas of Synthetic Adversarial Data Generation and the filtering and rewriting approaches have been adopted as standard practices for post-training LLMs, including Cohere's Command R+ model. These methods improve general robustness across diverse user query types.

Additionally, the concept of adversarial human evaluation has been further explored in the context of LLM safety through the now well-established practice of red teaming (Perez et al., 2022; Ganguli et al., 2022). Red teaming has evolved into an import concept in LLM safety that involves dedicated teams or individuals assuming adversarial roles to identify and measure potential risks and vulnerabilities in LLMs. By adopting an adversarial mindset, red teams can proactively uncover harmful or unethical behaviours exhibited by these models. This approach has become a key component for enhancing LLM safety and mitigating potential harm to

users and society.

A dynamic approach to data collection, where models are iteratively trained on collected data and then used in-the-loop for the next round of data collection, has become a common practice in developing LLMs. This approach enhances the effectiveness of the collected data and the models' performance. Furthermore, model-in-the-loop evaluation has become the norm for assessing Language Models, particularly when measuring the general capabilities of these systems across a broad spectrum of tasks.

Chapter 6

Generative Annotation Assistants

Synthetic Adversarial Data Generation (SADG), as introduced in the previous chapter, can be used to improve both the general performance and robustness of models. It involves the curated creation of synthetic data that mimics the strategies a human might employ to identify model failure modes. This approach is particularly valuable when aiming to maximise model robustness without the need for additional human annotation or costly data collection campaigns. SADG offers a cost-effective way to push model performance and robustness beyond what existing data can offer, making it a valuable tool in the model development and refinement process.

The research question then arises: *Are the question-generation models used in Synthetic Adversarial Data Generation constrained by the space probed by humans in the initial adversarial datasets and can we drive further improvements by assisting humans in constructing adversarial examples?*

In this chapter, we explore innovative ways to support human adversaries. By developing tools and methodologies to assist humans, it becomes easier to uncover adversarial examples and edge cases that might otherwise be challenging to identify. This collaboration between humans and Generative Annotation Assistants (GAAs) can lead to a more comprehensive understanding of model behaviours and potential weaknesses. Assisting humans in this manner not only improves the overall efficiency of the process, allowing humans to find viable adversarial examples faster and cheaper, but also the effectiveness, improving the ability of human annotators to identify queries that the models struggle with.

Having humans and generative assistants collaborate on the same task allows both the benefits of rapid automated exploration provided by the assistants and the creativity and context awareness of humans to work together to make the adversarial data collection process more cost effective and higher quality, ultimately also leading to downstream performance and robustness improvements.

This work explores a broad range of ideas and techniques for providing humans with effective assistance, including different generative mode training setups, sampling strategies, and answer generation and filtering strategies, and further provides insight into how annotator and GAA interactions play out.

This material discussed in this chapter is based on the published work titled *"Models in the Loop: Aiding Crowdworkers with Generative Annotation Assistants"* authored by **Max Bartolo**, Tristan Thrush, Sebastian Riedel, Pontus Stenetorp, Robin Jia and Douwe Kiela.

This work was published and presented at the 2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) in Seattle, Washington. Additional material from the First Workshop on Dynamic Adversarial Data Collection (DADC) held at the same conference is also referenced in the end-of-chapter reflection.

6.1 Overview

In Dynamic Adversarial Data Collection (DADC), human annotators are tasked with finding examples that models struggle to predict correctly. Models trained on DADC-collected training data have been shown to be more robust in adversarial and out-of-domain settings, and are considerably harder for humans to fool. However, DADC is more time-consuming than traditional data collection and thus more costly per annotated example. In this work, we examine whether we can maintain the advantages of DADC, without incurring the additional cost. To that end, we introduce Generative Annotation Assistants (GAAs), generator-in-the-loop models that provide real-time suggestions that annotators can either approve, modify, or reject entirely. We collect training datasets in twenty experimental settings and perform

Figure 6.1: Example interaction between an annotator and the models in the loop. The annotator selects an answer from the passage, for which the Generative Annotation Assistant (GAA) prompts a question. The annotator can then freely modify the question and/or answer, or generate another prompt. In the adversarial data collection setting, a model-in-the-loop provides predictions with the aim of encouraging annotators to find model-fooling examples. In the answer prompting setting, an answer suggestion is prompted by the assistive model instead of being selected by the annotator.

a detailed analysis of this approach for the task of extractive question answering (QA) for both standard and adversarial data collection. We demonstrate that GAAs provide significant efficiency benefits with over a 30% annotation speed-up, while leading to over a 5x improvement in model fooling rates. In addition, we find that using GAA-assisted training data leads to higher downstream model performance on a variety of question answering tasks over adversarial data collection.

6.2 Introduction

Natural language processing has become increasingly reliant on large datasets obtained using crowd sourcing. However, crowdsourcing as an unconstrained annotation approach is known to result in machine-exploitable annotator artefacts (Jia

and Liang, 2017; Schwartz et al., 2017; Gururangan et al., 2018; Geva et al., 2019), leading to poor out-of-distribution generalisation (Chen et al., 2016; Weissenborn et al., 2017; Yogatama et al., 2019; McCoy et al., 2019). Dynamic Adversarial Data Collection (DADC) aims to address these issues by introducing state-of-the-art models into the data collection loop and asking human annotators to produce examples that these models find challenging (Kiela et al., 2021). The intuition behind this approach is that it leads human annotators to better explore the space of possible examples. Previous work has found that DADC leads to improved model robustness on adversarial datasets (Nie et al., 2020; Bartolo et al., 2020), increased sample diversity (Bartolo et al., 2020; Wallace et al., 2022), better training data (Wallace et al., 2022) and better domain generalisation (Bartolo et al., 2021).

Despite these advantages, a downside to DADC is that it increases the human effort necessary to annotate a single example and thus the overall annotation cost. In fact, to date, only a limited number of large-scale training datasets have been produced using DADC and its application has been primarily restricted to producing challenge sets or as additional training data to improve the performance of models already trained on non-DADC curated datasets. To make better use of DADC data, Chapter 5 proposes generating synthetic adversarial training sets to further improve model robustness. However, this approach inevitably limits example diversity as it relies on examples ultimately generated by a model with no additional human input, and provides no guarantees that useful synthetic examples would transfer across target adversary models of varying capabilities or across annotation rounds.

In this work, we propose assisting annotators by having generative models aid human annotators in the data collection loop. Concretely, we utilise a Generative Annotation Assistant (GAA) model that provides prompt suggestions to crowdworkers, while allowing full flexibility for edits and rewrites to support example generation while still allowing for human creativity as shown in Figure 6.1. We explore GAAs in a broad range of experimental settings, including standard and adversarial data collection approaches, training on various source datasets, and employing sampling methodologies based on likelihood, adversarial feedback, and un-

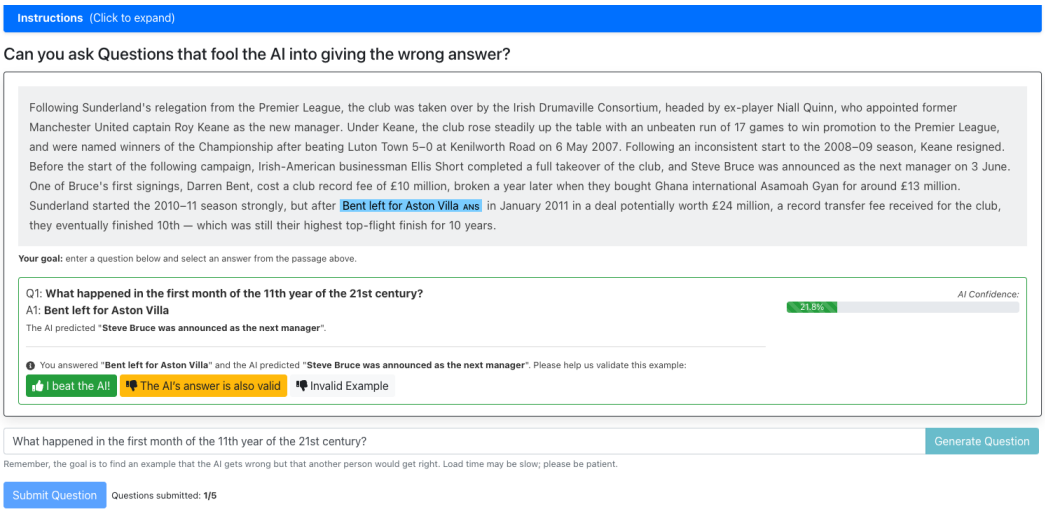


Figure 6.2: The Annotation Interface used for data collection. This example shows a question generated using a generative assistant trained on the AdversarialQA data and selected an adversarial sampler, which successfully allowed the annotator to beat the QA model in the loop.

certainty. We showcase the value of this approach on the task of extractive question answering (QA), and find that GAAs can help improve both the standard and adversarial data collection paradigms. We find considerable efficiency gains, with over 30% observed annotation speed-ups, as well as improved data effectiveness with up to a 6.1F_1 improvement in downstream performance over adversarial data collection.

6.3 Related Work

6.3.1 Dynamic Adversarial Data Collection (DADC)

There is a rich body of recent work showing the benefits of dynamic adversarial data collection in model evaluation (Yang et al., 2018b; Dua et al., 2019; Dinan et al., 2019; Nie et al., 2020; Bartolo et al., 2020; Kiela et al., 2021; Wallace et al., 2022), although the approach has been challenged for not necessarily leading to better generalisation on non-adversarial test sets (Kaushik et al., 2021) and being sensitive to the choice of model that was used in the loop (Bowman and Dahl, 2021; Phang et al., 2022). This work builds on previous work in adversarial data collection methods for QA (Bartolo et al., 2020), and work investigating the use

of question generation models to create synthetic adversarial data to improve QA model robustness (Bartolo et al., 2021).

6.3.2 Generative Model Annotation Support

A long line of prior work has trained generative models for question answering (Du et al., 2017; Du and Cardie, 2018; Zhao et al., 2018; Lewis and Fan, 2019; Alberti et al., 2019; Puri et al., 2020; Yang et al., 2020; Bartolo et al., 2021; Lewis et al., 2021b). In many cases, these approaches filter out questions that an external QA model gets wrong, in order to ensure correctness of the generated questions; our filtering strategies instead focus on generated questions that QA models get wrong as we hypothesise that these would serve as more useful initial prompts to human annotators.

Generative models have also been used to aid experts with writing contrast sets (Wu et al., 2021; Ross et al., 2022), but to the best of our knowledge, this is the first work to investigate the use of generative annotation assistants for crowdworkers directly in the annotation loop for NLP. Recent work on supporting crowdworkers for textual entailment in a non-adversarial setting shows no improvements on downstream transfer performance over baseline, albeit with reductions in previously observed issues with annotation artefacts (Bowman et al., 2020). Subsequent work highlights the need for further data collection efforts focusing on improving writing-based annotation processes (Vania et al., 2021), which we aim to investigate in this work. Separately, Ettinger et al. (2017) provide *breakers* with the ability to minimally edit original data to identify the boundaries of system capabilities, while Potts et al. (2021) analyse the use of prompts to assist crowdworkers in beating a model in the loop for sentiment analysis. In both cases, prompts are sourced from existing datasets and are not generated on the fly.

6.3.3 Active Learning and Weak Supervision

Active learning approaches have been used to accelerate annotation (Tsuruoka et al., 2008), although this typically assumes access to a pool or stream of unlabelled data for which the learning algorithm can query labels (Settles, 2009). In our setting,

no unlabelled questions are provided, necessitating the use of a generative model to suggest questions instead. Moreover, our annotators are free to edit and browse generated questions, whereas annotators in active learning typically only provide labels and have no choice in what to label. Some of our sampling and filtering strategies based on entropy are inspired by uncertainty sampling, a standard active learning algorithm (Lewis and Gale, 1994).

6.4 Experimental Setup

Our study focuses on the effects of incorporating generative annotation assistants, and understanding their interactions with annotators and discriminative models-in-the-loop in a DADC context for QA. We provide crowdworkers with a short passage from Wikipedia and ask them to write five questions and highlight the span in the passage that best answers the question for each (see Figure 6.2). We pay workers equally across experiment modes to avoid creating an incentive imbalance and pay out an additional bonus for each question that successfully beats the discriminative QA model i.e., for each question that the model fails to answer correctly. Finally, we validate all collected examples using a separate worker pool that also undergoes rigorous onboarding and validation. We ask three of these additional workers to report on the validity of each annotated example.

Selected Passages We select passages from KILT (Petroni et al., 2021) to allow for the possibility of future investigation into cross-domain and task transfer. We restrict KILT passages to those with between 100 and 600 tokens that are used by at least 5 of the KILT tasks. Furthermore, we filter out any passages with any 8-gram overlap (after normalisation) to the SQuAD1.1 training or development sets, seeking to ensure that all passages used in our study are novel and previously unseen by the discriminative QA models in the loop. This leaves a total of 10,109 passages from 421 Wikipedia pages. We retain and supply all passage-relevant KILT metadata (such as IDs and provenances) with our collected datasets to facilitate future work.

Model-in-the-Loop The discriminative QA model in the loop is $\text{ELECTRA}_{\text{Large}}$ (Clark et al., 2020) trained on SQuAD1.1 and AdversarialQA, and enhanced using SynQA to improve adversarial robustness as investigated in Chapter 5.[1] This model represents the best-performing model on the Dynabench (Kiela et al., 2021) leaderboard at the time of conducting this study, obtaining a word-overlap F_1 score of 94.5% on the SQuAD1.1 dev set, and represents the state-of-the-art on AdversarialQA achieving 77.6% on the $\mathcal{D}_{\text{BiDAF}}$ subset, 71.5% on $\mathcal{D}_{\text{BERT}}$, and 63.2% on $\mathcal{D}_{\text{RoBERTa}}$.

Generator-in-the-Loop For our generative models, we use the *fairseq* (Ott et al., 2019) implementation of $\text{BART}_{\text{Large}}$ (Lewis et al., 2020), and fine-tune the decoder to generate questions conditioned on the passage and the answer highlighted by the annotator. To provide annotators with a diverse set of questions, we decode using nucleus sampling with $top_p = 0.75$, as decoding using standard beam search results in questions which are more similar to each other and therefore likely to be less useful as question prompts to annotators. To speed up inference and model-annotator interaction, we preemptively identify answer candidates for each passage and generate questions to build up a large cache from which we serve questions during annotation. Once there are no questions remaining in the cache for a particular answer, or if the annotator selects an answer that is not in the cache, we fall back to querying the generative model in real-time. In this work, we investigate generative assistants trained on three different sources of questions: SQuAD1.1, AdversarialQA, and the combination of both SQuAD and AdversarialQA.

Question Sampling We investigate three different selection strategies for presenting the generated questions as prompts to annotators: i) *generator likelihood* samples candidates in the order prescribed by the generative model's associated likelihood values; ii) *adversarial sampling* selects generated questions in order of the least word-overlap F_1 scores when queried against the discriminative QA model; and iii) *uncertainty sampling* is inspired by active learning and selects generated questions in order of the least span selection confidence when queried against the QA model. The latter two provide an interesting trade-off for exploration as we

[1] You can interact with this model at https://dynabench.org/models/109.

would expect the quality of the generated questions to be worse than if sampled based on likelihood. However, we hope that such prompts could serve to inspire annotators and provide a "starting point" beyond the answering capabilities of the QA model, irrespective of correctness. We hypothesise that modifying such examples might be a more effective process for annotators to undertake than when starting from higher quality but less model-confusing prompts, and investigate this question thoroughly.

Answer Prompts We also investigate the effects of abstracting away the answer selection task from the annotator. To identify potential candidate answers, we use Self-Attention Labelling (SAL) (Bartolo et al., 2021) and investigate providing annotators with both answer prompts as well as the corresponding generated questions.

Experimental Settings In total, there are twenty different experimental settings involving combinations of the above-mentioned pipeline components. We collect 2,000 validated training examples for each of these settings, for a total of 40,000 examples. For downstream evaluation we train $\text{ELECTRA}_{\text{Large}}$ QA models on the training datasets collected for each setting, and perform identical model selection and hyper-parameter tuning.

Annotation Interface We use an adaptation of the Dynabench (Kiela et al., 2021) QA interface that allows annotators to interact with the models in the loop, and further allows them to edit and modify generated questions and answers as required. The same base interface is used across experimental settings and only varied minimally depending on the current setting, for example by changing the title and instructions in the adversarial annotation setting, or by adding a "Generate Question" button when the setting involves GAAs. In the GAA settings, annotators are not informed what generative model they are interacting with, or what sampling mechanism is being used.

Crowdsourcing Protocol We use Amazon Mechanical Turk to recruit workers for this study and run all experiments using Mephisto.[2] To ensure proficiency in En-

[2] github.com/facebookresearch/Mephisto

Adversary-in-the-loop?	t (s)	vMER (%)	t/vMFE (s)	$\text{SQuAD}_{\text{dev}}$	$\mathcal{D}_{\text{BiDAF}}$	$\mathcal{D}_{\text{BERT}}$	$\mathcal{D}_{\text{RoBERTa}}$	MRQA
✗	**57.2** $_{23.9}$	0.63	11,537	**85.7**	43.5	28.3	21.1	52.0
✓	61.5 $_{27.1}$	**1.86**	**4,863**	85.0	**53.0**	**34.2**	**26.9**	**58.8**

Table 6.1: Baseline results comparing standard and adversarial data collection. *t* shows the median time taken per example in seconds and median absolute deviation (subscript). *vMER* is the validated model error rate. *t/vMFE* is the time per validated model-fooling example. Lower is better for the time-dependent metrics. Downstream evaluation is measured by training an $\text{ELECTRA}_{\text{Large}}$ QA model on the collected datasets and evaluating F_1 scores on the SQuAD1.1 dev set, the AdversarialQA test sets, and the MRQA dev sets for domain generalisation.

glish, crowdworkers are required to be based in Canada, the UK, or the US. They are also required to have a Human Intelligence Task (HIT) Approval Rate greater than 98%, have previously completed at least 1,000 HITs, and undergo a dedicated onboarding process. Workers were randomly assigned to one of the possible experiment modes and were all presented with passages sampled from the same set, for which they were tasked with writing and answering five questions. All collected questions were than validated for correctness by a separate group of crowdworkers. We collect three validations per question and use this information, along with manual verification of a subset of the annotated examples, to maintain a high level of quality and remove examples from workers with less than an 80% validity rate. We calculate the reliability of agreement between validators using Fleiss' kappa at 0.46. Workers were provided an additional bonus for each example validated as having successfully fooled the model in the adversarial data collection settings. In total, 1,931 workers participated in the study, with 1,559 contributing to the final datasets. We also continuously validate both annotators and validators based on signals such as repetitiveness, agreement, and manual checks.

Evaluation We evaluate the outcomes in each of the experimental settings by a selection of metrics:

i. *median time per example* as a measure of annotation efficiency and where a lower time taken is better;

ii. *validated Model Error Rate (vMER)* (Bartolo et al., 2021) which evaluates the effectiveness of annotators at generating valid question-answer pairs that

Sampling Strategy	t (s)	vMER (%)	t/vMFE (s)	$\text{SQuAD}_{\text{dev}}$	$\mathcal{D}_{\text{BiDAF}}$	$\mathcal{D}_{\text{BERT}}$	$\mathcal{D}_{\text{RoBERTa}}$	MRQA
Likelihood	$\mathbf{37.5}_{21.4}$	0.62	8,708	**85.5**	41.8	25.3	20.3	53.6
Adversarial	$55.6_{20.9}$	**4.02**	**1,760**	84.7	**45.5**	26.1	20.0	**54.3**
Uncertainty	$63.1_{28.6}$	2.77	3,018	83.2	**45.5**	**28.2**	**21.9**	53.2

Table 6.2: Results for the investigation into supporting standard data collection using GAAs. Since this setting assumes no access to adversarially-sourced data, we use a generative model trained only on questions from SQuAD1.1. There is no adversarial QA model in the loop in this setting.

the QA model in the loop fails to answer correctly;

iii. *median time per validated model-fooling example* which serves as a single metric incorporating both method efficiency and effectiveness and thus provides a convenient metric for comparison across the various experimental settings; and

iv. *downstream effectiveness* in which we evaluate the performance (by word-overlap F_1 score) of a QA model trained on the data collected in each of the experimental modes on the standard SQuAD1.1 benchmark, on the AdversarialQA benchmark, and in terms of domain generalisation ability on the MRQA (Fisch et al., 2019) dev sets.

Lower values are better for the time-dependent metrics, however, from the perspective of training data we consider a higher vMER to be better guided by the performance benefits observed for adversarial over standard data collection. This is corroborated by comparison with downstream results.

6.5 Results

Our study allows us to perform a thorough investigation into both the efficiency and effectiveness of the different data annotation methodologies. It also allows us to build on work investigating the various differences between standard and adversarial data collection (Kaushik et al., 2021).

6.5.1 Standard versus Adversarial Data Collection

The standard and adversarial data collection settings we use as baselines do not make use of GAAs, and are designed to replicate the SQuAD1.1 (Rajpurkar et al.,

2016) and AdversarialQA (Bartolo et al., 2020) annotation setups as closely as possible. However, in contrast to AdversarialQA, our setting only provides annotators with a financial incentive to *try* to beat the model in the loop through the use of a bonus, and does not restrict annotators to only submitting model-fooling examples.

The results, shown in Table 6.1, highlight the differences between the two annotation approaches. As expected, standard data collection is more efficient in terms of the time taken per example, as there is no requirement for annotators to make any effort to try to beat a model. However, the efficiency differences are not as large as seen in settings where annotators *have* to submit model-fooling examples (Bartolo et al., 2020).

We also find considerable benefits from adversarial data collection in terms of the validated model error rate and subsequent downstream performance. As observed in Chapter 3, adversarial data collection is more effective on adversarial test sets and aids domain generalisation, with slight performance degradation in the standard evaluation setting, which may be mitigated by increasing the amount of training data or combining with non-adversarial training data (for detailed results, refer to Appendix C.2). The combined performance across evaluation settings is considerably higher for adversarial data collection.

We note that the training data sizes in both these experimental settings are relatively small, and the benefits of adversarial data collection have been shown to be more pronounced in the low data regime, likely due to increased example diversity. Furthermore, while the passages used in this study are sourced from Wikipedia, there may exist characteristic differences between these and the passages used in SQuAD.

We also observe considerably lower (i.e., better) adversarial human evaluation vMER scores achieved for our synthetically-augmented ELECTRA$_{\text{Large}}$ model-in-the-loop compared to the 8.8% reported for RoBERTa$_{\text{Large}}$ in Chapter 5. We hypothesise that this is primarily due to two factors: the improved robustness of ELECTRA in comparison to RoBERTa, and more tightly-controlled example validation. For further evidence of the improved adversarial robustness of ELECTRA, refer to

GAA Training	Sampling	t (s)	vMER (%)	t/vMFE (s)	$\text{SQuAD}_{\text{dev}}$	$\mathcal{D}_{\text{BiDAF}}$	$\mathcal{D}_{\text{BERT}}$	$\mathcal{D}_{\text{RoBERTa}}$	MRQA
SQuAD	*Likelihood*	$61.1_{31.9}$	2.25	3,501	**86.5**	50.1	30.1	24.1	57.6
SQuAD	*Adversarial*	$62.6_{22.7}$	4.66	1,750	83.2	48.1	27.7	24.2	55.0
SQuAD	*Uncertainty*	$62.1_{26.3}$	2.41	3,317	86.1	**51.8**	31.1	24.4	**58.4**
AdversarialQA	*Likelihood*	$54.2_{24.4}$	3.09	2,458	84.7	49.8	36.9	**29.8**	56.8
AdversarialQA	*Adversarial*	$63.9_{25.6}$	**6.30**	**1,262**	83.3	49.4	34.9	28.1	56.7
AdversarialQA	*Uncertainty*	$72.1_{30.9}$	5.20	1,776	83.6	50.3	**37.3**	27.0	55.7
Combined	*Likelihood*	$\mathbf{53.6}_{25.6}$	2.64	2,724	85.1	48.9	33.4	24.7	56.5
Combined	*Adversarial*	$69.6_{29.6}$	4.86	1,922	82.9	48.0	33.6	28.6	54.5
Combined	*Uncertainty*	$62.0_{25.0}$	4.87	1,690	85.3	50.5	33.9	28.8	56.5

Table 6.3: Results for the investigation into supporting adversarial data collection using GAAs. We investigate three different GAA training dataset sources, and three sampling strategies. The adversarial QA model used in the annotation loop is identical for all settings.

Appendix C.3.

6.5.2 Improving Standard Data Collection

We now investigate whether it might be possible to improve standard data collection practices using generative assistants – *can we achieve similar performance to adversarial data collection without access to any adversarial data?*

We therefore use a GAA trained on SQuAD1.1, and investigate the three sampling techniques namely: likelihood, adversarial, and uncertainty sampling. Results are shown in Table 6.2. We find that using a GAA with likelihood sampling considerably improves the efficiency of the annotation process in comparison to the standard data collection baseline in Table 6.1. It also gives comparable vMER results and downstream QA performance.

Furthermore, both the adversarial and uncertainty sampling strategies prove effective. While the reduction in time taken per example is not as substantial as for standard likelihood sampling, and is comparable to the standard data collection baseline, the vMER – an indicator of the diversity of the collected training data – is substantially improved and outperforms the adversarial data collection baseline. The downstream results are also promising, providing slight improvements over the standard data collection setting, particularly with regards to domain generalisation. They start to make progress towards the values obtained for the adversarial data collection baseline although, despite the improved vMER, overall downstream

GAA Training	Sampling	t (s)	vMER (%)	t/vMFE (s)	$\text{SQuAD}_{\text{dev}}$	$\mathcal{D}_{\text{BiDAF}}$	$\mathcal{D}_{\text{BERT}}$	$\mathcal{D}_{\text{RoBERTa}}$	MRQA
AdversarialQA	*Likelihood*	**38.5** $_{22.7}$	5.63	988	83.8	49.7	**40.3**	**30.7**	55.9
AdversarialQA	*Adversarial*	44.2 $_{21.6}$	9.46	**668**	83.9	48.7	36.2	30.3	55.2
AdversarialQA	*Uncertainty*	49.9 $_{24.8}$	7.80	854	84.8	**51.3**	38.9	30.6	56.3
Combined	*Likelihood*	45.9 $_{23.4}$	2.90	2,196	**85.2**	51.1	37.5	28.1	**56.4**
Combined	*Adversarial*	56.3 $_{27.1}$	**9.53**	785	83.5	48.4	35.5	29.3	55.5
Combined	*Uncertainty*	57.6 $_{27.7}$	4.48	1,841	83.4	47.4	36.6	27.8	55.2

Table 6.4: Results for the investigation into supporting adversarial data collection using GAAs equipped with answer prompting. We investigate two different GAA training dataset sources, and three sampling strategies. The adversarial QA model-in-the-loop is identical for all settings.

performance is considerably higher in the adversarial data collection setting.

In summary, these results shows that we can encourage annotators to come up with more challenging examples without requiring any adversarially-collected data or an adversarial model in the loop, simply through the use of GAAs paired with an appropriate sampling strategy. However, using adversarial data collection still provides substantially better downstream performance. These observations are in line with our initial hypothesis that sampling generated prompts from regions of known model uncertainty, or prompts that we know the model finds challenging to answer, irrespective of generated sample quality, provides annotators with a better starting point for example creation.

6.5.3 Improving Adversarial Data Collection

Following the efficiency gains observed for standard data collection, we investigate whether it is possible for GAAs to provide further improvements over adversarial data collection. As for the previous experiments, we investigate GAAs trained on three different datasets: SQuAD1.1, AdversarialQA, and the combination of both. We combine each of these with the three previously discussed sampling strategies resulting in nine different experimental settings. Results are shown in Table 6.3.

We find that when annotators are incentivised to try to beat an adversarial QA model-in-the-loop, the previously seen efficiency gains are not as clear cut. In fact, annotators are slightly slower than for the adversarial data collection baseline when using a SQuAD-trained GAA. When using a GAA that has been trained on adversarially-sourced questions, standard likelihood sampling provides efficiency

gains over the baseline, however, both adversarial and uncertainty sampling (which naturally lead to more complex prompts that might be more challenging to work with) actually slow annotators down, although they do provide improved validated model error rates and overall better adversarial example generation efficiency measured by the time taken per validated model-fooling example.

In terms of downstream performance, there is no clear best option, but the best settings consistently outperform the adversarial data collection baseline on the most challenging examples ($\mathcal{D}_{\text{BERT}}$ and $\mathcal{D}_{\text{RoBERTa}}$) while providing comparable results in the other evaluation settings. Surprisingly, we find that various settings, particularly those involving a SQuAD-trained GAA can provide performance gains over the standard data collection baseline on SQuAD1.1. We also observe that a SQuAD-trained GAA with uncertainty sampling gives best performance on the less challenging evaluation sets, while an AdversarialQA-trained GAA gives best performance on the evaluation datasets collected using a more performant adversary. This is also in line with the observations made in Chapter 3 showing a distributional shift in question type and complexity with an increasingly stronger model-in-the-loop.

The general takeaway in terms of the ideal experimental setting from the perspective of downstream performance is that it depends on the particular evaluation setting, with GAAs trained on examples from a particular setting yielding better performance when the downstream model is also evaluated in similar conditions. Another key observation is that both the validated model error rate and time per validated model-fooling example comfortably outperform the baselines across the board, highlighting the enhancements to the effectiveness of the annotation process provided by incorporating GAAs in the loop.

6.5.4 Investigating Answer Prompting

The settings explored in the previous sections focus on investigating the effects of assisting free-text generation of the questions using GAAs. However, the QA crowdsourcing setting also involves annotation of answer spans, which we also explore in the search for efficiency gains. Here, we explore GAAs trained on datasets

with adversarially-sourced components and the same three sampling strategies as previously (likelihood, uncertainty and adversarial), while additionally providing annotators with an answer suggestion.

In essence, this is similar to an answer and question validation setting, with the difference that annotators have the ability to freely modify both answer and question, or request additional suggestions. Results for our experiments involving answer assistance are shown in Table 6.4.

We find that answer prompting is very effective at improving annotation efficiency, providing gains in all six experimental settings while also providing improved vMER results in all cases. We also see very similar downstream performance result patterns to the previous set of experiments – for performance on the more challenging evaluation sets ($\mathcal{D}_{\text{BERT}}$ and $\mathcal{D}_{\text{RoBERTa}}$), an AdversarialQA-trained GAA with likelihood sampling gives best performance, while for performance on SQuAD, a GAA trained on examples including SQuAD gives the best results. As previously discussed and as shown in Appendix C.2, adding SQuAD examples to the training data mitigates this effect.

The consistency in performance patterns serves to further highlight the previous observation that, while using GAAs provides considerable gains in both the efficiency of the annotation process and effectiveness in terms of downstream results, the ideal annotation setup should be selected based on the target downstream evaluation. It is also worth highlighting the considerable performance improvements on the more challenging AdversarialQA evaluation sets observed when using an AdversarialQA-trained GAA even over adversarial data collection.

6.6 Annotator Interaction with GAAs

While we provide annotators with instructions explaining how they can use the GAAs to aid their annotation, they are free to query the generative models as many times as they like, if at all, during annotation. We are interested to see how the three main factors affecting interaction with the GAAs that we explore – training data, sampling strategy, and answer prompting – affect the ways in which annotators

Feature	Setting	Avg. #Generations per Example
GAA Training	SQuAD	0.69
	AdversarialQA	0.86
	Combined	0.83
Sampling	*Likelihood*	0.67
	Adversarial	0.87
	Uncertainty	0.83
Answer Prompt?	✗	0.73
	✓	0.91

Table 6.5: Results showing how often annotators query the GAA for different experimental settings.

interact or use the GAAs.

Results, shown in Table 6.5, indicate that annotators query the GAA less frequently when being shown simpler prompts i.e. those obtained using a GAA trained on non-adversarially sourced examples, or selected using likelihood sampling which tends to provide higher quality and less complex generated texts. We also find that annotators query the GAA more frequently when an answer prompt is also provided. We believe that this can be attributed to the fact that the answer and question prompt setting is more similar to a validation workflow, allowing annotators to generate prompts until a satisfactory one is found.

6.7 Discussion and Conclusion

In this work, we introduce Generative Annotation Assistants (GAAs) and investigate their potential to aid crowdworkers with creating more effective training data more efficiently. We perform a thorough analysis of how GAAs can be used for improving QA dataset annotation in different settings, including different generative model training data, sampling strategies, and whether to also provide annotators with answer suggestions.

We find that GAAs are beneficial in both the standard and adversarial data collection settings. In the standard data collection setting, and under the assumption of no access to adversarially-collected data, GAAs with prompts sampled based

on likelihood provide annotation speed-ups, while prompts sampled by adversarial performance or uncertainty metrics provide benefits to both the model error rates on the collected data as well as subsequent downstream QA performance. We find that while GAAs are effective for improving standard data collection, we still do not approach the performance obtained when using adversarial data collection.

For adversarial data collection, we demonstrate improved effectiveness of the annotation process over the non-GAA baseline, although this comes at a cost of reduced annotation efficiency. We show that also aiding annotators with answer prompts boosts data collection efficiency even beyond that of standard data collection, while retaining overall downstream performance.

We find that the ideal annotation setting differs for different intended evaluations, with an uncertainty-sampled GAA trained on data that was not adversarially-collected providing best performance on simpler questions, while a GAA trained on adversarially-collected data provides best downstream performance on more challenging evaluation sets. However, we also find that combining with a small sample of SQuAD training examples can boost performance on these less-challenging questions, and that in this setting a likelihood-sampled adversarially-trained GAA consistently provides the best results.

In terms of efficiency, we see annotation speed-ups over baseline of 34.4% for standard data collection and 37.4% for adversarial data collection. In terms of effectiveness, we see over a 5x improvement in vMER over adversarial data collection, along with downstream performance gains. We improve over standard data collection on $\text{SQuAD}_{\text{dev}}$ by up to 0.8F_1 and improve over adversarial data collection by up to 6.1F_1 on $\mathcal{D}_{\text{BERT}}$, and 3.8F_1 on $\mathcal{D}_{\text{RoBERTa}}$. Furthermore, we see benefits in domain generalisation over standard data collection, and show that annotators interact with the GAA more frequently when it has been trained on adversarially-collected data, is sampled based on adversarial or uncertainty feedback, and also provides answer prompts.

While our analysis is limited by the size of the collected data, we believe that GAAs can help drive further innovation into improved data collection methodolo-

gies based on these findings. We hope that our analysis of various aspects of GAA incorporation into the annotation pipeline and the interactions between annotators and multiple models in the loop can help inform future work exploring broader aspects of GAA use, such as for other NLP tasks or for larger scale annotation efforts.

6.8 Reflection

This chapter provides an in-depth exploration of methods for assisting human annotators with writing adversarial questions through the use of Generative Annotation Assistants (GAAs), finding some effective setups and others that are considerably more effective. We demonstrate over 30% efficiency improvements for both standard and adversarial data collection with the use of GAAs, an over 5x improvement in validated Model Error Rate (vMER) over the baseline in the adversarial setting, and improved downstream performance *and robustness* when models are trained on data collected in this way.

Crowdworkers succeed at finding valid adversarial examples (against models that are already particularly robust) around 2% of the time when incentivised to do so through instructions, bonuses, and other means. Our best GAA setups increase this score considerably, up to ~10%, but what does this tell us about the abilities of human adversaries to identify and probe model blind spots and failure modes?

The first time that the work in Chapter 3 was presented externally, an esteemed professor at Cambridge University asked a particularly intriguing question: *"Have we reached the limit of what annotators are able to contribute?"*. My response, along the lines of us getting closer to what can be achieved, at least in a crowdworker-based annotation setting, suggested that there was more to explore. This question, coupled with the desire to explore just how far we could push things, provided part of the motivation for the research presented in the preceding chapters. While the results were impressive, we sought to dig deeper into the implications on the limits of human adversary performance in a broader context.

We organised the First Workshop on Dynamic Adversarial Data Collection at NAACL '22 in Seattle, Washington (Bartolo et al., 2022a). Along with the in-

credible speakers, panels, and networking events exploring and discussing various topics around dynamic adversarial data collection, we also hosted a Shared Task – a competition that teams from various academic and industrial institutions could participate in. The DADC Shared Task focused on Extractive Question Answering (QA) as a reference task, along the same lines as the work presented in this thesis. It was hosted on Dynabench (see Chapter 4) and further broken down into two competitive tracks:

Track 1: Better Annotators. Participants were required to submit question answering (QA) examples through the Dynabench platform, to form part of the evaluation set for Track 2. The objective was to find as many model-fooling examples as possible – the primary incentive was to be the team with the highest validated model error rate (vMER). Submitted examples were then validated by the task organisers.

Track 2: Better Training Data. This was a data-centric track focused on participants selecting or curating the best selection of 10,000 training examples from existing standard or adversarial datasets (see Chapter 3), additional expert-annotation or crowdsourcing (see Chapter 4), or synthetic-generation (see Chapters 5 and the work described above).

The *Better Annotators* track saw teams develop different strategies and approaches in an attempt to beat the AI, exploring aspects such as individual and collaborative ideation, self-validation, using linguistically informed attack strategies (such as garden-path sentences[3], and complex co-reference resolution), non-linguistically informed reasoning skills (such as numeric manipulation or list manipulation), and taking advantage of model bias and priors.

The winning team, team *longhorns*, a group of faculty, postdocs, and students from the UT Austin linguistics department, computer science department, information school, and electrical and computer engineering department – essentially highly qualified *expert annotators*, achieved a remarkable 65% validated Model Error Rate (vMER), likely influenced by their systematic, linguistically-informed, and collaborative approach (Kovatchev et al., 2022). The second-placed team, team *fireworks*,

[3] A classic example, often attributed to Thomas Bever, is "The horse raced past the barn fell."

also managed a very respectable 55%. These scores are particularly remarkable in the context of the thesis author also providing a symbolic baseline submission and, having a reasonable level of expertise developing work in this space coupled with deep understanding of the requirements of the task having led its design and organisation it, achieving a vMER score of 33%.

These results, along with those from other participating teams, highlight the gap between even highly trained and qualified, strongly incentivised, assisted crowdworkers (using GAAs) and expert annotators. It suggests that there is indeed a practical benefit to involving expert annotators in the adversarial data collection process, at least from the perspective of attack success rates.

The data collected through the DADC Shared Task is available for future research and also includes expert-authored failure mode explanations – a new feature added for this task, which could be used to train critique models, enhanced reward models, or for explanation-informed evaluation.

It is also worth noting the overlap between the techniques and strategies explored by the participants of the DADC Shared Task and many of the challenges being faced with the development of contemporary LLMs. Many of the innovations and techniques explored both in the work presented in this chapter and by participants in the DADC Shared Task, continue do inform research into improving and robustifying LLM development for a broad, and often generic, set of capabilities across a range of tasks and domains.

Chapter 7

Conclusions

This thesis has explored a series of challenges around Language Model robustness, investigating approaches and interactions between humans and models in the data annotation loop to understand, probe and improve adversarial robustness.

After introducing the work and providing relevant background in Chapters 1 and 2, we have investigated the effects of human annotators interacting with models-in-the-loop to identify, probe, and ultimately correct for model failure modes in Chapter 3. To paraphrase task feedback from one of the study participants: "This task is fun and extremely challenging, but we are actively and directly contributing to the improvement of these models. The AIs are already really strong, but I feel like we are contributing to our own demise."

In Chapter 4, we introduced Dynabench, a research platform for dynamic adversarial data collection and benchmarking, and expand on this with the introduction of two novel concepts; Synthetic Adversarial Data Generation (SADG) in Chapter 5 and Generative Annotation Assistants (GAAs) in Chapter 6, complimentary methodologies for further improving model robustness.

7.1 Summary of Contributions

We provide a summary of the key findings and contributions presented in this thesis, organised into thematic groups for clarity.

7.1.1 Community Resources

This work introduced a number of dataset, model and software contributions made publicly available to the community for use and further development.

AdversarialQA, introduced in Chapter 3 is among the most-downloaded Question Answering datasets on the Hugging Face hub with over 1 million total downloads. It has contributed to improving the performance and robustness of various academic and enterprise systems (Ye et al., 2022; Jang et al., 2023), including many state-of-the-art Large Language Models (Sanh et al., 2022; Chung et al., 2024), and has also influenced thinking around evaluation and understanding of system performance (Wang et al., 2021; Khashabi et al., 2022; Paranjape et al., 2022; Perez et al., 2023; Lee et al., 2023).

Dynabench is an open-source research platform for dynamic data collection and benchmarking released under an MIT License, with source code available at https://github.com/mlcommons/dynabench. The platform is also currently hosted and maintained by the DMLR working group of MLCommons at https://dynabench.org/. Dynabench has contributed to and influenced a large sub-field of research in the space, including around the development of datasets and resources, modelling methodologies and evaluation, as described in detail in Chapter 4. Dynabench continues to support research efforts with collaborators from academic and industrial institutions including University College London, ETH Zurich, the University of Oxford, the University of Birmingham, the University of Maryland, Stanford University, Harvard University, Meta, Google, Coactive, NASA, Common Crawl, and Cohere.

SynQA, introduced in Chapter 5, is a dataset of over 300k synthetically generated adversarial examples, made available under an MIT license at https://huggingface.co/datasets/mbartolo/synQA.

Question Answering Models. Throughout this work, we make various Extractive Question Answering models available as community resources. These models are described in Chapter 5 and are available on the Hugging Face model hub at https://huggingface.co/mbartolo/roberta-large-

synqa, https://huggingface.co/mbartolo/roberta-large-synqa-ext, and https://huggingface.co/mbartolo/electra-large-synqa. They have been downloaded nearly 20k times, with RoBERTa-Large-SynQA-Ext being the most popular model, and have offered valuable contributions to both academic research and practical, real-world applications.

Question Generation Models. Question Generation models used in Chapters 5 and 6 have also been made available at https://github.com/maxbartolo/synQA-question-generators, along with a supporting codebase and instructions for use. They have also been used to support various other research experiments and investigations (Lewis et al., 2021b; Paranjape et al., 2022; Bartolo et al., 2022a).

7.1.2 Dynamic and Adversarial Approaches

Two themes consistently revisited throughout this work are the *dynamic* and *adversarial* aspects of data collected both for model training and evaluation.

Dynamic. The theme of *dynamic* data collection is first touched upon in Chapter 3, where we explored the effects of using different models, with different architectures pre-trained in different ways and of different performance and general capability levels, in-the-loop. This highlighted some findings of particular interest including that; i) data collected using weaker models was highly beneficial for training stronger models, ii) data collected using strong models posed more challenging evaluation settings, iii) individual models are susceptible to blind spots, which can be overcome to some degree through adversarial training, and iv) the best setting from a performance perspective was to combine data collected against different models in the loop. This focus on dynamic data collection was further extended directly through Dynabench in Chapter 4, but also indirectly through the subsequent work where the datasets and models developed were used to power subsequent rounds of the Dynabench Question Answering Task. For example, the previously described DADC Shared Task involved human expert annotators attempting to beat models that had been trained and developed based on the work described in Chapters 3 and 5. This *dynamic* theme continues to play a critical role in the devel-

opment of contemporary state-of-the-art language models, played out on a longer timeframe, and it has become a conventional approach to data collection with the standard Learning from Human Feedback (LHF) setting involving a pool of candidate models, typically at least two, sampled for completion generation given a provided prompt. Human annotators then identify which of the completions they prefer, and this preference data is used to re-train models on a regular cadence, which are then used to replace the previous models in-the-loop. A similar approach is often taken when red teaming language models, to allow safety aspects related to model characteristics to improve as a function of the dynamic nature of the model probing and data collection process (Touvron et al., 2023).

Adversarial. The *adversarial* theme is central to and permeates throughout the presented work. We build on earlier work that extended the concept of adversarial examples, typically minor non-semantic perturbations to an input that cause an undesirable and often significant change in the output, to the human-in-the-loop setting where humans interact with models to probe model blind spots and create human-written adversarial examples. This human-and-model-in-the-loop approach to creating adversarial examples allows humans to use their creativity to explore the space of model strengths and weaknesses while relying on direct and immediate feedback to form an intuition around effective strategies. The data collected in this manner is highly valuable both for adversarial training, in which models are trained on adversarially-collected data for general performance and robustness improvements, as well as evaluation on challenging examples. It is worth noting that in the adversarial training setup, our experimental results indicate that training solely on adversarially-collected data is not the objective, rather a mixture of examples collected with and without models-in-the-loop, and ideally with a diverse selection of models, provides the best overall performance. We further develop and investigate these ideas around *adversarial* data collection throughout this thesis, showing that human-written adversarially-collected training data can help improve both in- and out- of domain performance. Throughout this work, we have explored various aspects surrounding adversarial interaction and its implications, from investigating

the ways humans and machines interact in adversarial settings, generating synthetic adversarial data, and assisting human annotators in the creation of adversarial examples using specifically trained generative models. Furthermore, this work introduced Adversarial Human Evaluation (AHE) which has inspired a large body of work around testing model performance and capabilities across a broad range of capabilities and dimensions, and has become an essential part of the process for developing contemporary LLMs (Perez et al., 2022; Ganguli et al., 2022).

7.1.3 Humans and Models In-the-Loop

The broader theme of involving humans and models in-the-loop, goes back at least to the dawn of modern computing where, in a test of intelligence now commonly referred to as the *Turing test*, the assumed best strategy for a machine to prove its capability to think was to "try to provide answers that would naturally be given by a man" (Turing, 1950).

This thesis explored various dimensions of the human-and-model-in-the-loop paradigm, including with multiple models as in the self-training stages of the Synthetic Adversarial Data Generation pipeline (see Chapter 5) and multiple humans collaborating to compete against question answering models as in the DADC Shared Task (see Chapter 6). This collaborative and competitive dynamic is also explored in a single environment in Chapter 6 where human annotators collaborate with generative question-writing assistive models in an attempt to fool other question answering models. Ideas around involving humans and models in the loop have now also become standard in the post-training of Large Language Models, particularly within the context of the Learning from Human Feedback (LHF) paradigm.

7.1.4 Improved Robustness

While we have explored various aspects of model performance, such as general performance on fixed standard and adversarially-collected test sets, domain generalisation, and capability-specific checklists, the central focus of this work has been on improving model robustness. The motivation for this, as alluded to in the introduction, is to narrow the gap between current natural language processing technologies

and their widespread application to create real-world value.

Improved robustness enhances the reliability and accuracy of the systems we build, making them better equipped to handle variations in input data, including noise, errors, and unexpected scenarios — essentially allowing them to perform as expected "in the wild", where data and user interactions can be messy and unpredictable. Robust systems are also more secure and reduce the potential for misuse or abuse. This is critical when considering the impact these models can have, especially in sensitive areas like content generation, language translation, or decision-making processes. Finally, robust systems create trust, providing users with consistent and accurate interactions, with systems failures occurring in ways that are predictable and easily mitigated, creating further opportunities for innovation and development of even more capable systems.

In this thesis, we have introduced and explored various techniques for improving model robustness, and have also provided evidence of the efficacy of these techniques in both academic and real-world settings.

7.1.5 Evaluation

Another theme we have touched on throughout this work is *evaluation*. A prerequisite for identifying and improving weaknesses in existing models is the ability to measure them. We have discussed various contributions and innovations around evaluation. In particular, in Chapter 3 we introduced three new adversarial test sets, the ground truth answers to which remain unreleased, making them a particularly valuable research resource. In Chapter 4 we discussed Dynabench, a research platform designed to support benchmarking and evaluation efforts, and discuss the impact it has had on the community. In Chapter 5 we introduced Adversarial Human Evaluation as part of the evaluation strategy for measuring robustness improvements resulting from Synthetic Adversarial Data Generation. We also briefly touched on the effects these ideas have had in the LLM space, with particular reference to *red teaming* as well as challenges around the current de facto standard for evaluating Large Language Models through human feedback (such as the LMSYS Chatbot Arena Leaderboard). This relatively simplistic approach is prone to being affected

by unintended confounders, and can under-represent critical aspects of model performance like factuality (Hosking et al., 2024).

Adversarially-curated evaluation datasets remain an effective technique for benchmarking the robustness capabilities of state-of-the-art LLMs. For example, the Llama3 Technical Report (Grattafiori et al., 2024) uses the AdversarialQA dataset introduced in Chapter 3 to test the models' question answering capabilities in challenging settings, suggesting that while non-adversarial capabilities improve in post-training, most of the performance on more challenging questions is acquired in pretraining. Furthermore, when evaluated on the AdversarialQA validation set in the 4-shot setting, Cohere's Command R+ Refresh (08-2024) model achieves 75.5% F_1 score on $\mathcal{D}_{\text{BiDAF}}$, 67.7% on $\mathcal{D}_{\text{BERT}}$, and 61.6% on $\mathcal{D}_{\text{RoBERTa}}$ giving a combined F_1 score of 68.2%. For reference, OpenAI's GPT4 (0613) model achieves a combined F_1 score of 64.3% while Meta's Llama 3.1 8B Instruct, 70B Instruct and 405B Instruct models achieve 57.8%, 69.0% and 68.9% respectively. While drawing a comparison between validation and test set performance is not ideal, these tend to fluctuate between approximately 1-2% of each other from previous experiments. The most competitive system we evaluate in this thesis, $\text{ELECTRA}_{\text{Large}}$ trained on AdversarialQA along with synthetic data, achieves a combined F_1 score of 69.4% (see Table B.8). While it is impressive that contemporary LLMs reach the same performance level as the highly-specialised models used in this work without dedicated finetuning, there remains headroom even to the combined non-expert lower-bound estimate on human performance of 74.4% (see Table 3.2), which widens with progressively more challenging questions such as on $\mathcal{D}_{\text{RoBERTa}}$.

7.2 Limitations

Throughout this thesis, we have acknowledged and addressed a range of limitations. As we conclude, we will revisit and summarise the key limitations for transparency and to support future research directions.

Task Diversity. In this work we have focused almost exclusively on Extractive Question Answering (also referred to as Reading Comprehension) as a reference

task. While various other tasks have been explored in the Dynamic Adversarial Data Collection paradigm including natural Language Inference (Nie et al., 2020), Sentiment Classification (Potts et al., 2021), Hate Speech Detection (Vidgen et al., 2021), and others, it is likely that the effects seen for a particular tasks are not identically felt across tasks, and it is as yet unclear what the magnitude of the benefit transfer to more general tasks is. The Dynabench platform, while highly flexible, has been limited to mostly text-only tasks with only a couple of vision-related tasks run so far, and could be extended to support various additional modalities as well as multilingual tasks.

Dataset Sizes. The sizes of the datasets presented, 36k examples for AdversarialQA and 300k+ for SynQA are modest, but not particularly large especially in the context of the scale of data that current systems are trained on. This represents a limitation in our ability to investigate the effects of scale.

Crowdworkers versus Experts. The majority of the work presented in this thesis focuses on the crowdsourced. While this is in itself, investigating and analysing the abilities of crowdworkers to collaborate and compete with machines, is an important contribution, it could have led to a ceiling effect on the complexity of the collected examples as alluded to and discussed with reference to small-scale expert annotation investigations in Chapter 6.

Participant Sizes: Human Evaluation. The participant sizes for the Adversarial Human Evaluation efforts described in Chapter 5 were relatively small. While the effect measured was sufficiently large, it is possible that increasing the human evaluator pool would result in a smaller robustness delta measurement. Furthermore, in Chapter 6, while 1,559 individuals contributed to the final dataset, this was split across 20 participant groups, with some groups having more or fewer participants than others.

Computational Resources. A serious limitation we have not yet addressed is the requirement for computational resources across all of the work described, including deployment and inference compute for the models in the loop, the considerable resources required to finetune the question answering and question generation models,

as well as the compute required to train and run the ablations presented across these studies. We aimed to limit our compute requirements to the minimum necessary wherever possible for sustainability and environmental reasons.

Applicability in the era of LLMs. Much of the research explored in this thesis has become part of the standard LLM development pipeline. Examples include: i) the high-level, long cycle human-and-model-in-the-loop iterative development process involving training an initial model, deploying it, allowing users to interact with the system and provide feedback on its failure modes, using both crowd-sourced and expert annotators to collect data addressing those failure modes and repeating, ii) dedicated adversarial human annotation efforts in the context of safety, nowadays referred to as red-teaming, iii) dynamic benchmarking which has become even more relevant giving the increasing rate at which benchmarks are saturating, and iv) synthetic data generation to maximise training signal. Other investigations, such as using generative models to assist in the annotation process, have seen some early adoption but we believe there is considerable room for exploration here. One of the primary limitations of this methodology is that it becomes increasingly more difficult to identify model failure modes as system capabilities improve. We observe this to some extent in this work, such as with the increasingly more capable models in Chapter 3 or the DADC Shared Task relying on adversarial expert annotators in Chapter 6. However, as we continue to scale this process, we should continue to innovate new approaches for maximising the error-correcting signal that models-and-humans-in-the-loop can provide. Furthermore, when we originally set out in this direction we were somewhat hopeful that adversarial training on data collected by humans directly probing for system failure modes would provide *total robustness*. While failure rates tend to drop significantly, we rarely see these reach zero. In many cases, it is still possible to adversarially cause a model to fail, even though it can be a lot more challenging. Our ultimate goal is still to guarantee a system's correctness for every valid input, even if on a set of relatively well-defined or constrained tasks. Whether an iterative failure-targeted approach will get us there or whether we need new approaches remains an open question.

7.3 Envisioning What's Next

We conclude by identifying broader areas for future work, and some trends we expect, or hope, to see in the future.

Towards Robust Models. Robustness has been a primary motivating factor for much of the work presented in this thesis. It remains a central tenet to the author's wider research agenda and a growing area of active research. Current research on large language model robustness is focused on improving the performance of models on downstream tasks. Robustness is multidimensional and captures various aspects of model performance that revolve around generalisation, adversarial resilience, and stability across diverse inputs and input perturbations. Recent work has looked at the impact of compression on the generalisability and robustness of pre-trained language models, with studies showing that compressed models are less robust than their counterparts on adversarial test sets (Du et al., 2023). Other research has explored the robustness of instruction-tuned LLMs to seen and unseen tasks, finding that performance worsens when dealing with unfamiliar instructions or instruction setups, such as in-context example ordering (Lu et al., 2022) or multiple-choice candidate answer ordering (Pezeshkpour and Hruschka, 2023; Zheng et al., 2024). Extending to the retrieval-augmented language model context, recent work explores methods for making models robust to irrelevant retrieved context while maintaining performance otherwise (Yoran et al., 2024). The increasing efforts to make models more robust contribute to making this technology more useful and, in consequence, accessible to society more broadly. This has further implications when extending beyond the text-only paradigm to the multimodal setting such as with images, video and speech.

Model Usability as a Feature. As the capabilities of the technologies being developed continue to evolve rapidly, the real-world value generated by such systems appears to be lagging behind their promise. A possible contributing factor is the field's focus on developing highly performant models from the perspective of high-level capabilities and skills, such as complex or mathematical reasoning, specialist domain knowledge, graduate-level knowledge processing, advanced data manip-

ulation, and providing language model agents access to use and execute various tools such as calculators or code interpreters. These are all admirable and exciting targets, and accomplishing any to any level of performance is impressive, but unless such tasks can be performed reliably and repeatedly, with low failure rates and predictable failure modes, there remain various obstacles between what the technology can offer and widespread adoption. There likely exist a range of tasks that current technologies could perform if they were designed-in, through user-focused features such as explainability and transparency providing clear justifications or critiques for model outputs enabling users to better interpret and rely on the LLM's responses (Lampinen et al., 2022; Turpin et al., 2023; Wu et al., 2023; Chen et al., 2024; Ye et al., 2024), multilingual support (Singh et al., 2024; Üstün et al., 2024; Aryabumi et al., 2024), personalisation and customisation through preamble editing, finetuning or techniques such as inverse constitutional AI (Findeis et al., 2024), improved error handling and feedback allowing the model to adapt and improve based on user feedback, integration with commonly-used tools and services such as productivity suites or communication workflows (Schick et al., 2023; Khattab et al., 2024), or calibrated model confidence scores to provide users with a sense of the model's confidence in the accuracy of a provided response (Kadavath et al., 2022).

Collaborative Competition. As LLMs and AI technologies become increasingly integrated into our daily lives, we hope to see the interactions between humans and machines continue to evolve and advance in various ways. In particular, we expect to see varying aspects of collaborative and competitive approaches, as well as innovative combinations of the two in ways that will eventually appear seamless and invisible to end users. We anticipate continued development in the directions pioneered in this work, including further exploration into using models as part of the feedback loop. This could involve employing models to identify failure modes and drive improvements, as well as assisting human users in the processes of enhancing and leveraging these models more effectively. This can be achieved through model-aided data creation, curation, or selection, or by providing assistance in using the models, for example through automatic prompt engineering or preamble customi-

sation to align with specific requirements or adapt to provided feedback (Shin et al., 2020; Fernando et al., 2024; Xian et al., 2024). Ultimately, we envision a future where such technology is part of our everyday lives, augmenting human effort in ways that are supportive and beneficial, with humans providing direct feedback, implicit, or corrective signal where the technology falls short of expectations, creating a tight yet subtle iterative improvement loop. The large context lengths available to current models, typically in the hundreds of thousands to millions of tokens (for reference, the BART models used in this work had a default context length of 1,024 tokens), further opens up new possibilities for incorporating feedback without having to retrain or finetune the model, by simply providing the feedback in-context. Furthermore, the interaction signal used to improve these models, currently captured in data, will become more direct as such systems become further embedded in our working environments. Ultimately, these systems will be working hand in hand with us, adapting to and learning from us and their environment on the fly, to better serve the needs of the tasks at hand.

Granular and Comprehensive Representation of Human Preference. Evaluation informs modelling progress, yet current approaches are limited. They tend to either focus on specific, highly curated task collections (Srivastava et al., 2023), typically based on "traditional" NLP tasks like question answering (Hendrycks et al., 2021a), or operate in an unconstrained setting where humans can freely query models at any level of complexity on any topic, providing a single feedback rating intended to encompass all the nuance of human preference, such as the LMSYS Chatbot Arena. We hope to see continued innovation in model evaluation, with increasing focus on dynamic, human-and-model-in-the-loop approaches, aiming to capture the multidimensional range of system performance (Hosking et al., 2024).

To quote Turing's work that introduced this thesis, *"we can only see a short distance ahead, but we can see plenty there that needs to be done"*.

Appendix A

Beat the AI

A.1 Additional Dataset Statistics

Question statistics In Figure A.2 we analyse question lengths across SQuAD1.1 and compare them to questions constructed with different models in the annotation loop. While the mean of the distributions is similar, there is more question length variability when using a model in the loop. We also perform analysis of question types by *wh-* word as described earlier (see Figure A.1). This is in further detail displayed using sunburst plots of the first three question tokens for $\mathcal{D}_{\text{SQuAD}}$ (cf. Figure A.6), $\mathcal{D}_{\text{BiDAF}}$ (cf. Figure A.8), $\mathcal{D}_{\text{BERT}}$ (cf. Figure A.7) and $\mathcal{D}_{\text{RoBERTa}}$ (cf. Figure A.9). We observe a general trend towards more diverse questions with increasing model-in-the-loop strength.

Answer statistics Figure A.4 allows for further analysis of answer lengths across datasets. We observe that answers for all datasets constructed with a model in the loop tend to be longer than in SQuAD. There is furthermore a trend of increasing answer length and variability with increasing model-in-the-loop strength. We show an analysis of answer types in Figure A.3.

A.2 Annotation Interface Details

We have three key steps in the dataset construction process: i) training and qualification, ii) "Beat the AI" annotation and iii) answer validation.

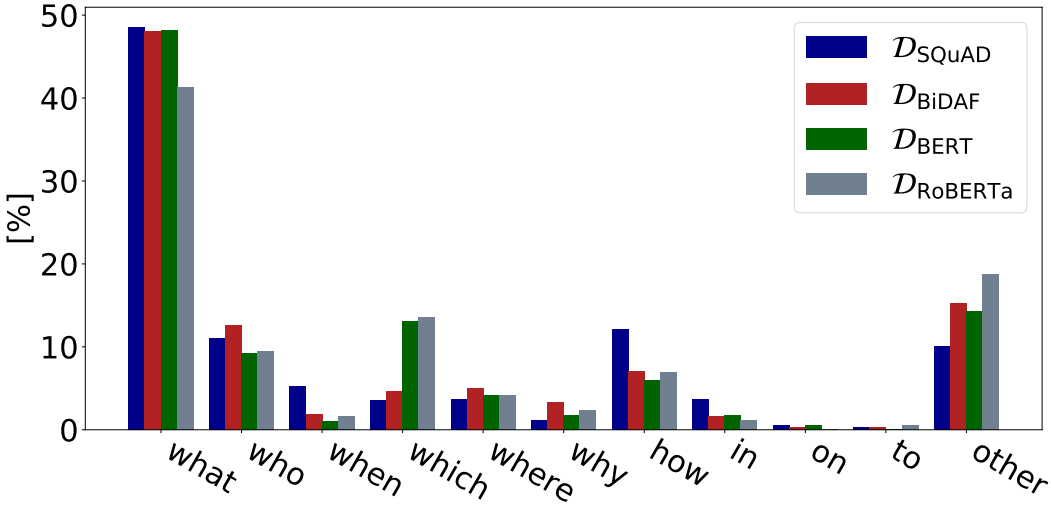


Figure A.1: Analysis of question types across datasets.

Training and Qualification This is a combined training and qualification task; a screenshot of the interface is shown in Figure A.10. The first step involves a set of five assignments requiring the worker to demonstrate an ability to generate questions and indicate answers by highlighting the corresponding spans in the passage. Once complete, the worker is shown a sample "Beat the AI" HIT for a pre-determined passage which helps facilitate manual validation. In earlier experiments, these two steps were presented as separate interfaces, however, this created a bottleneck between the two layers of qualification and slowed down annotation considerably. In total, 1,386 workers completed this task with 752 being assigned the qualification.

"Beat the AI Annotation" The "Beat the AI" question generation HIT presents workers with a randomly selected passage from SQuAD1.1, about which workers are expected to generate questions and provide answers. This data is sent to the corresponding model-in-the-loop API running on AWS infrastructure and primarily consisting of a load balancer and a *t2.xlarge* EC2 instance with the *T2/T3 Unlimited* setting enabled to allow high sustained CPU performance during annotation runs. The model API returns a prediction which is scored against the worker's answer to determine whether the worker has successfully managed to "beat" the model. Only questions which the model fails to answer are considered valid; a screenshot for

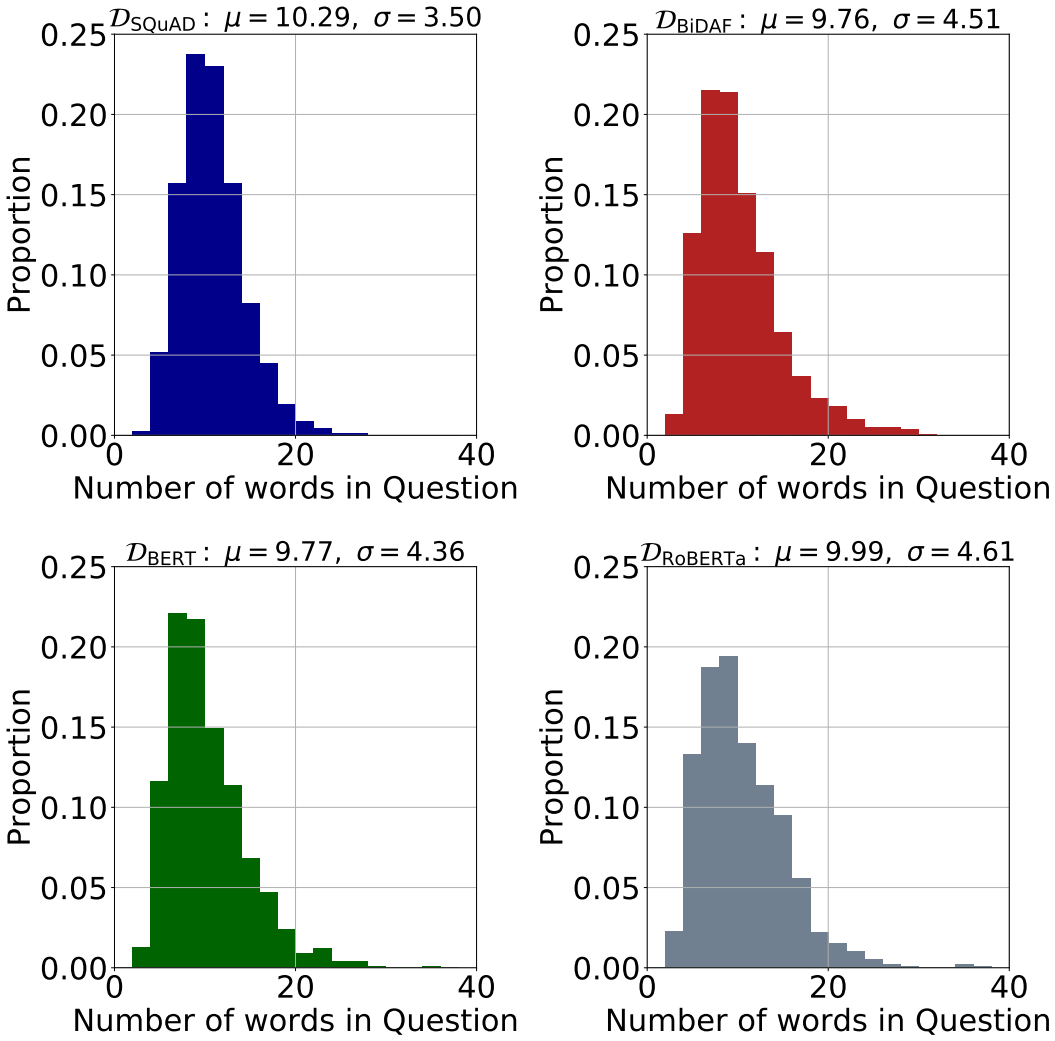


Figure A.2: Question length distribution across datasets.

this interface is shown in Figure A.11. Workers are tasked to ideally submit at least three valid questions, however fewer are also accepted – in particular for very short passages. A sample of each worker's HITs is manually validated; those who do not satisfy the question quality requirements have their qualification revoked and all their annotated data discarded. This was the case for 99 workers. Worker validation distributions are shown in Figure A.5.

Answer Validation The answer validation interface (cf. Figure A.12) is used to validate the answerability of the validation and test sets for each different model used in the annotation loop. Every previously collected question generation HIT from these dataset parts, which had not been discarded during manual validation,

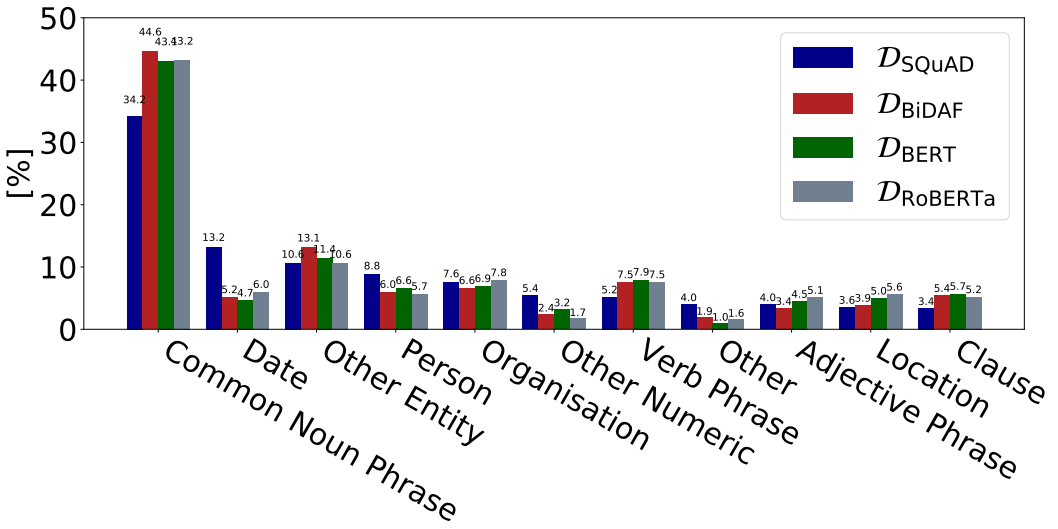


Figure A.3: Analysis of answer types across datasets.

is submitted to at least 3 distinct annotators. Workers are shown the passage and previously generated questions and are asked to highlight the answer in the passage. In a post-processing step, only questions with at least 1 valid matching answer out of 3 are finally retained.

A.3 Catalogue of Comprehension Requirements

We give a description for each of the items in our catalogue of comprehension requirements in Table A.1, accompanied with an example for illustration. These are the labels used for the qualitative analysis performed in Section 3.6.

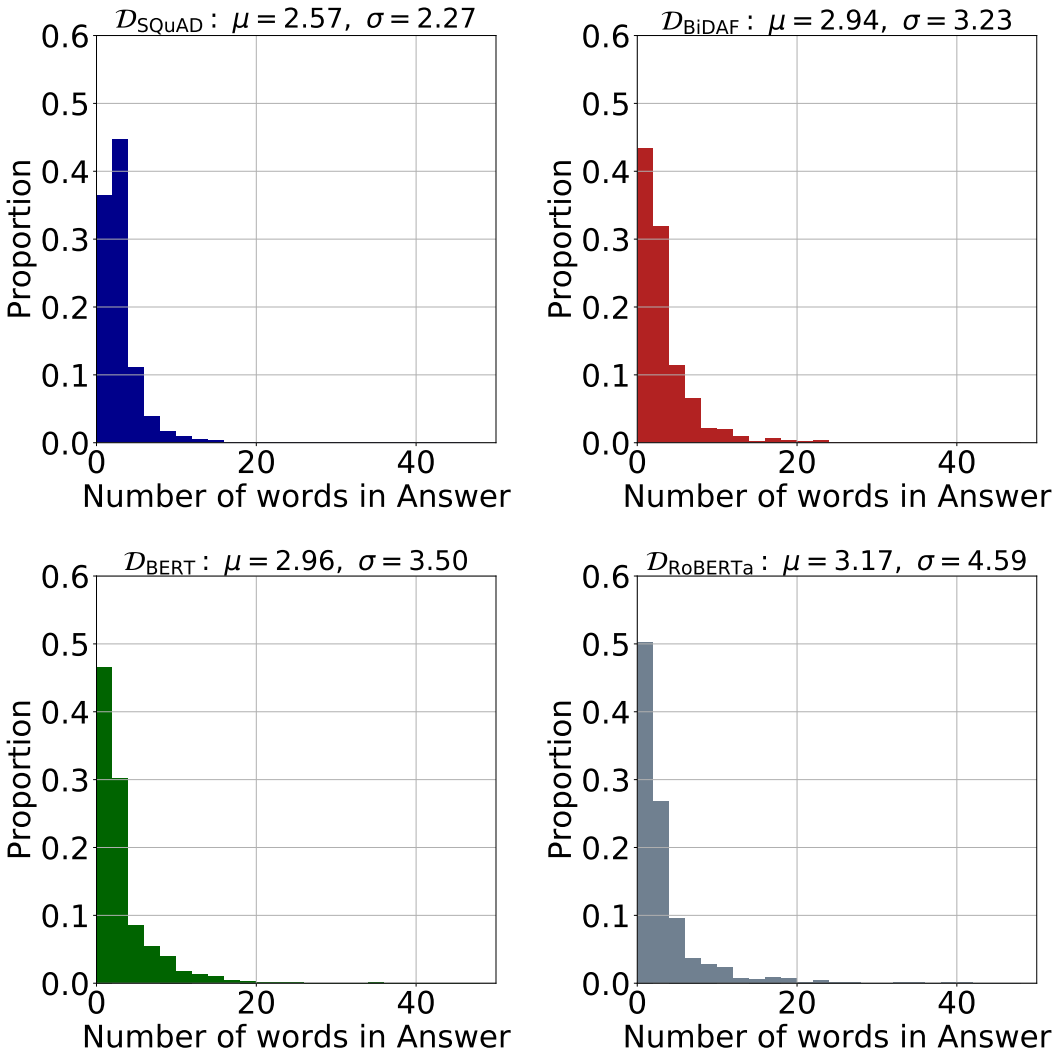


Figure A.4: Answer length distribution across datasets.

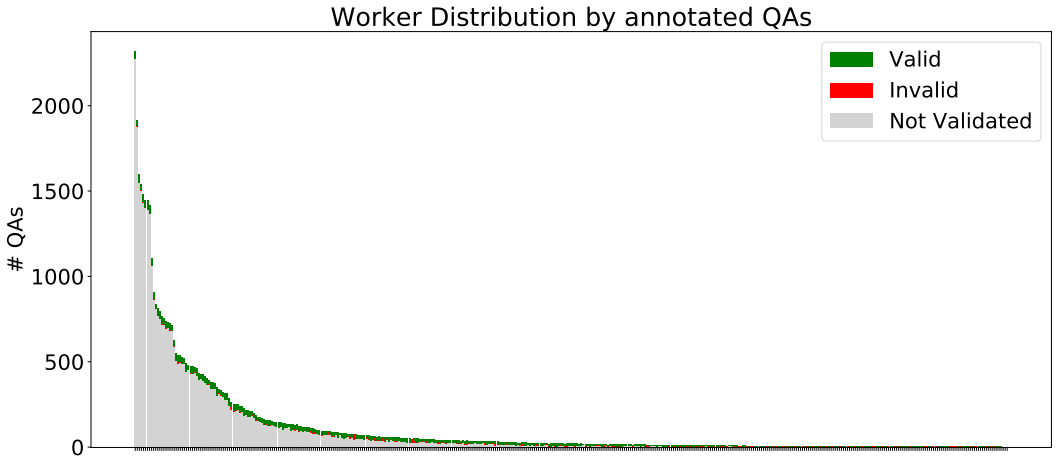


Figure A.5: Worker distribution, together with the number of manually validated QA pairs per worker.

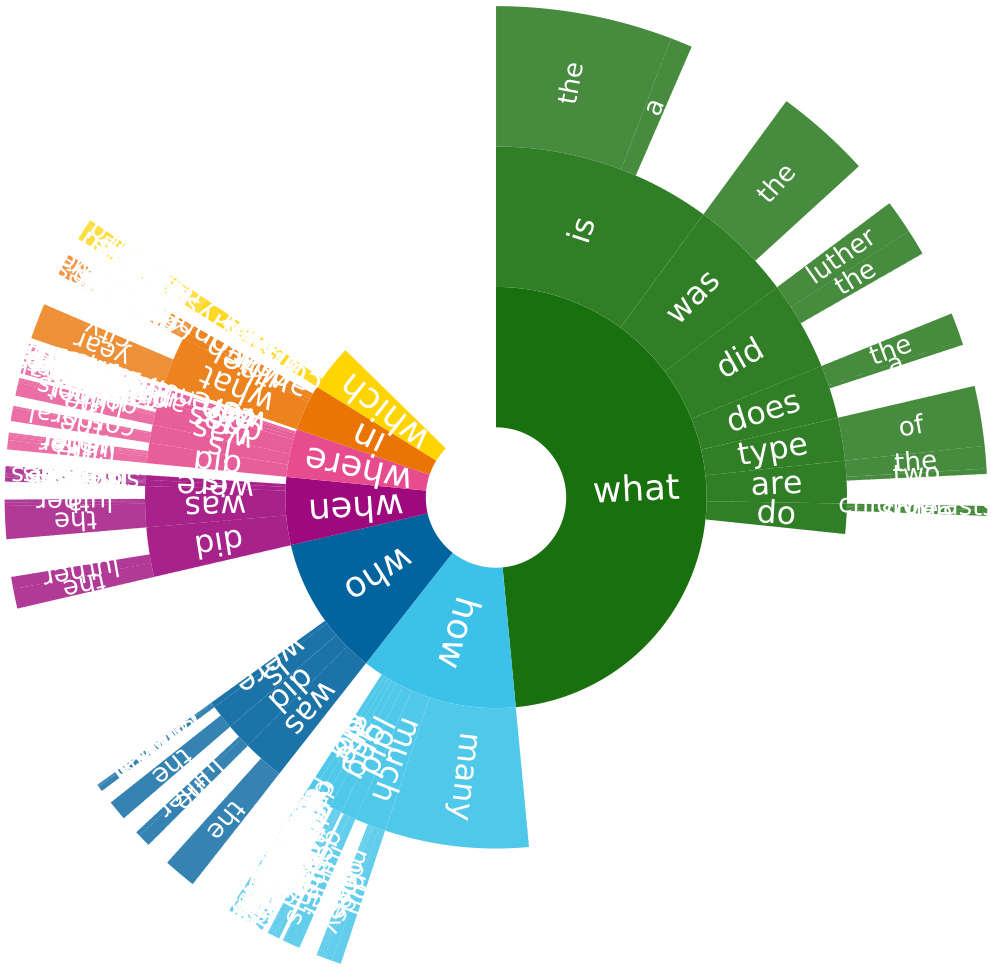


Figure A.6: Question sunburst plot for $\mathcal{D}_{\text{SQuAD}}$.

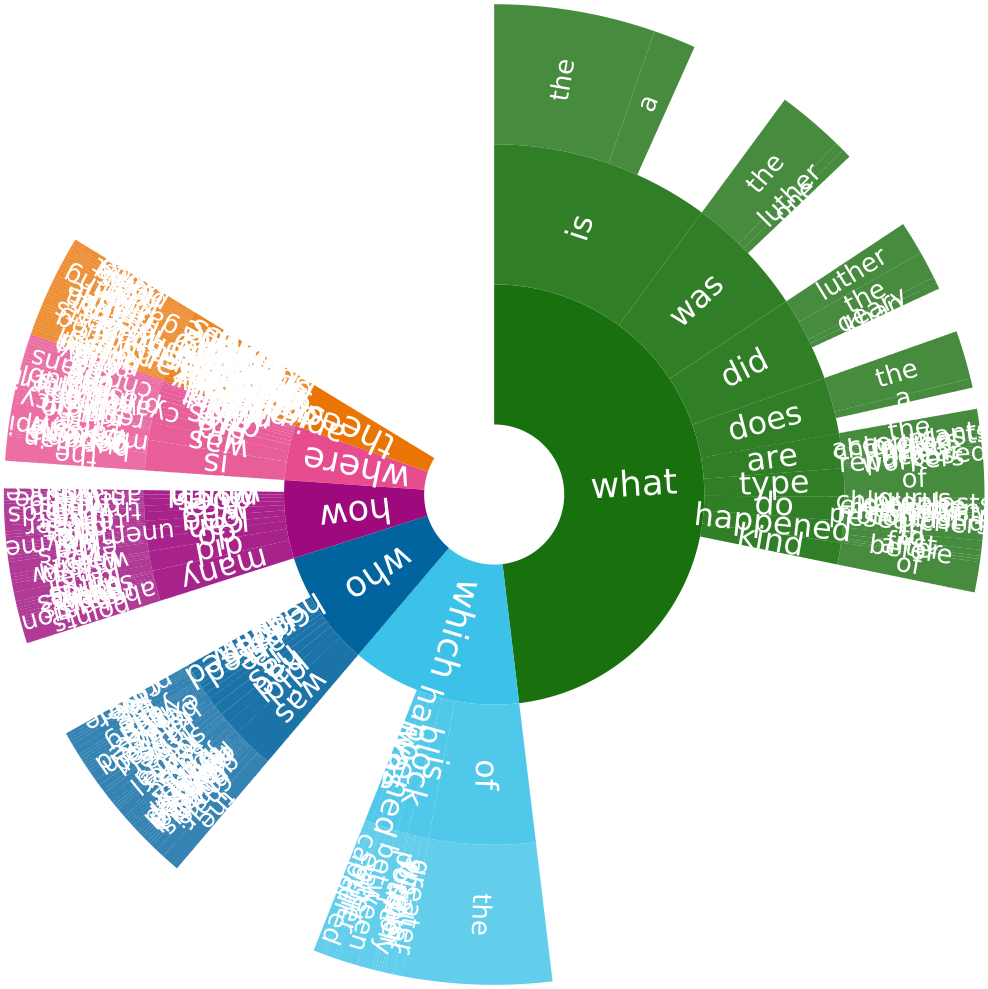


Figure A.7: Question sunburst plot for $\mathcal{D}_{\text{BERT}}$.

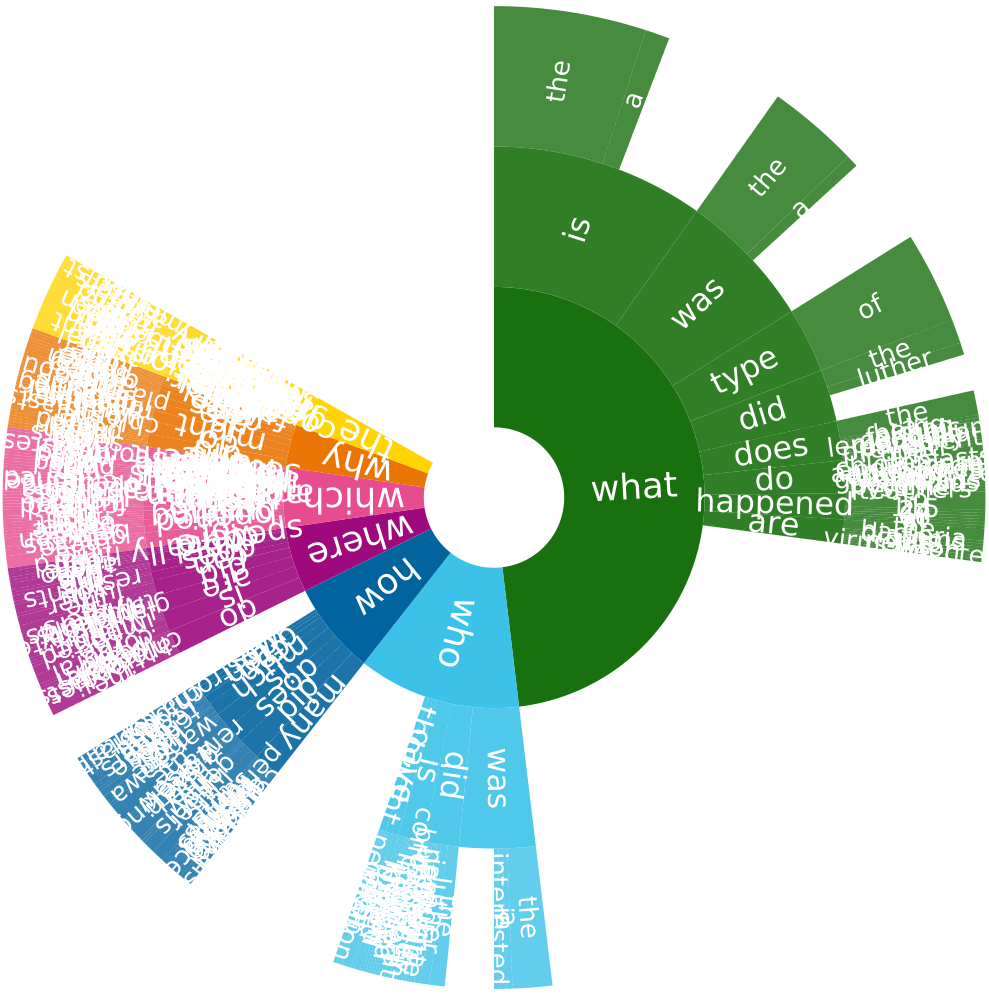


Figure A.8: Question sunburst plot for $\mathcal{D}_{\text{BiDAF}}$.

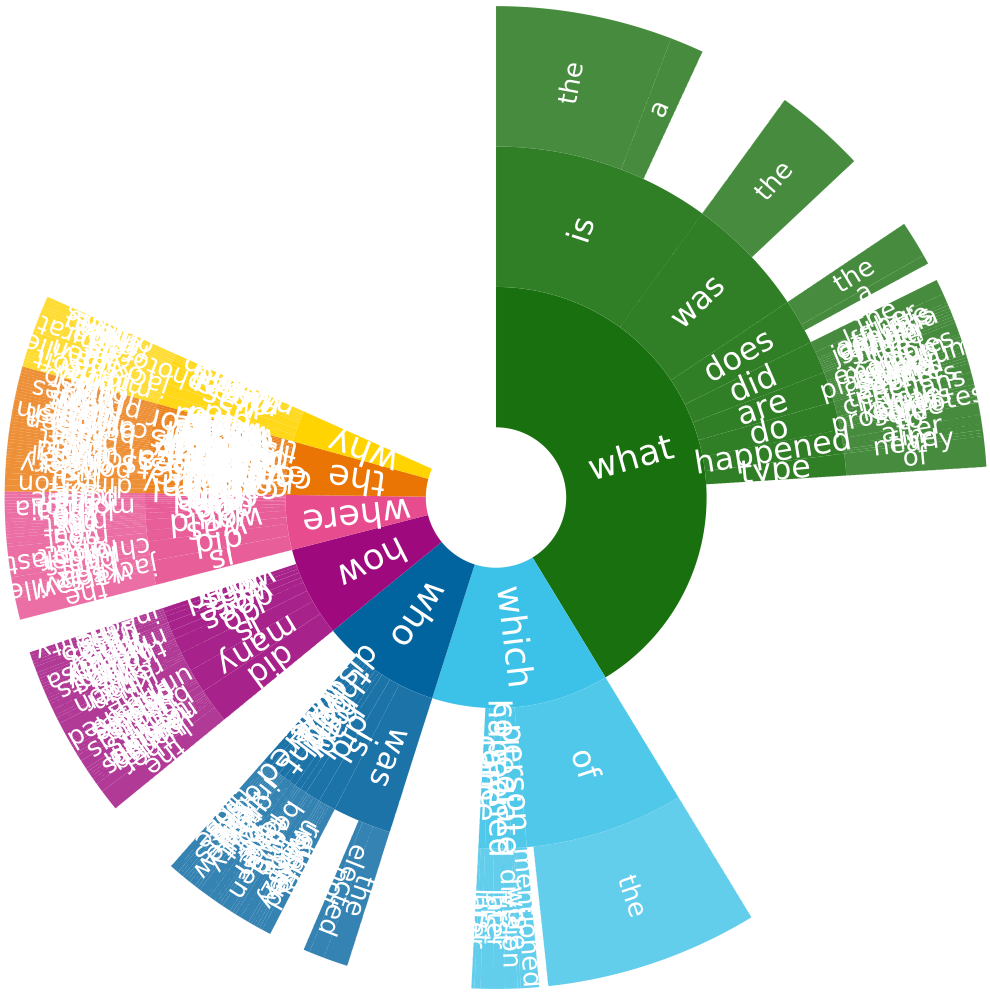


Figure A.9: Question sunburst plot for $\mathcal{D}_{\text{RoBERTa}}$.

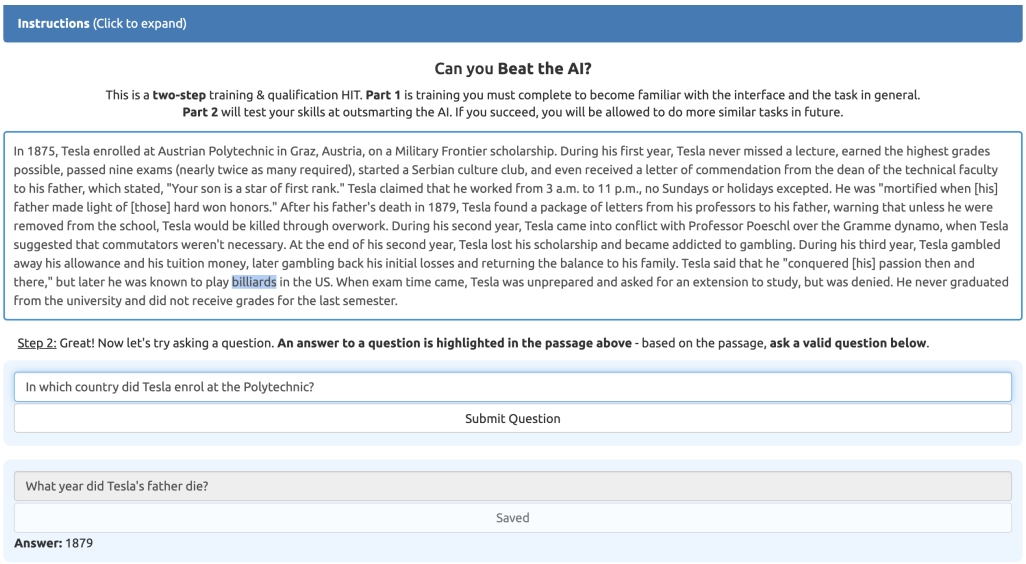


Figure A.10: Training and qualification interface. Workers are first expected to familiarise themselves with the interface and them complete a sample "Beat the AI" task for validation.

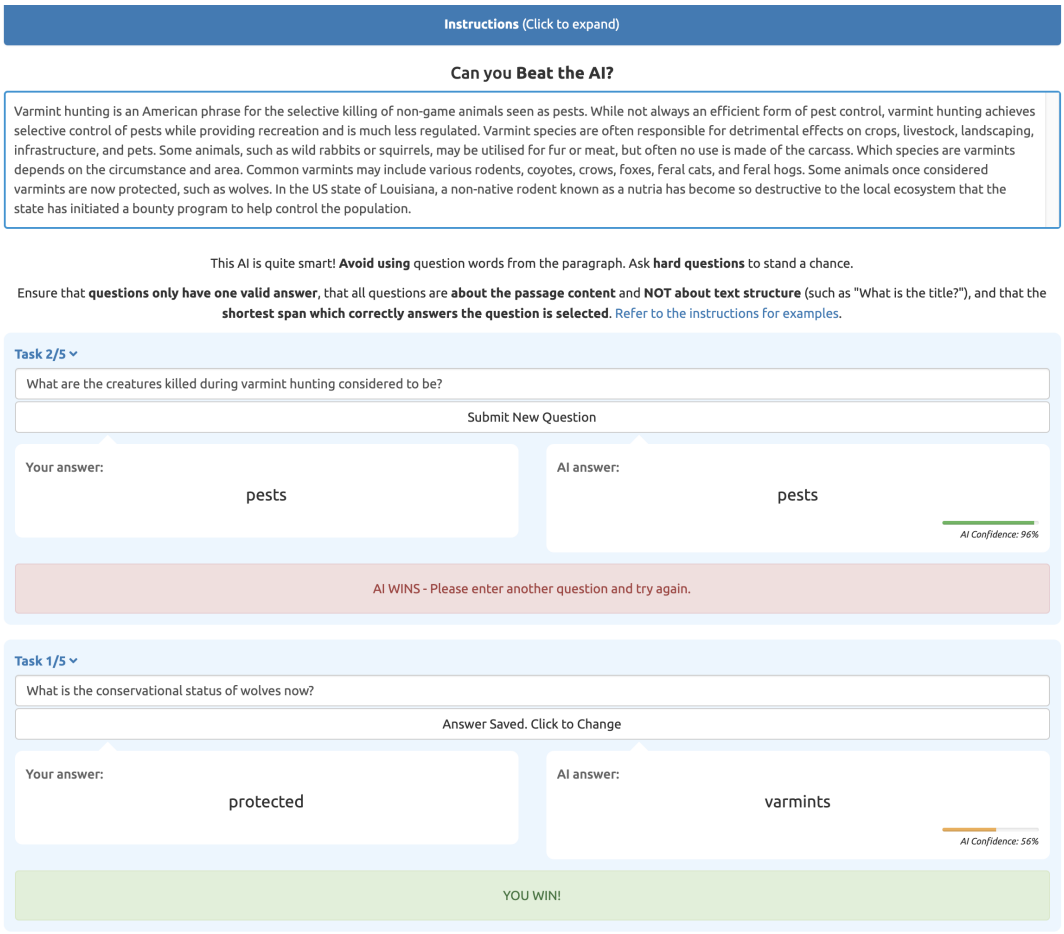


Figure A.11: "Beat the AI" question generation interface. Human annotators are tasked with asking questions about a provided passage which the model in the loop fails to answer correctly.

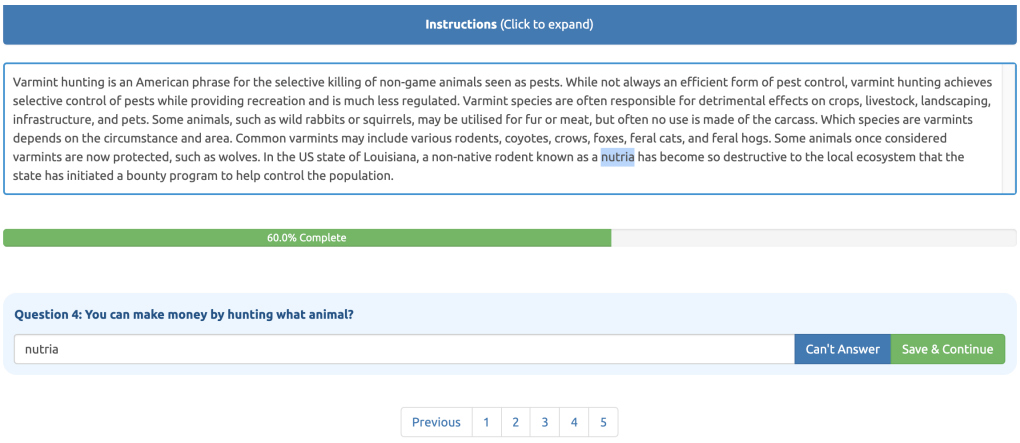


Figure A.12: Answer validation interface. Workers are expected to provide answers to questions generated in the "Beat the AI" task. The additional answers are used to determine question answerability and non-expert human performance.

Type	Description	Passage	Question
Explicit	Answer stated nearly word-for-word in the passage as it is in the question.	Sayyid Abul Ala Maududi was an important early twentieth-century figure in the Islamic revival in India [...]	Who was an important early figure in the Islamic revival in India?
Paraphrasing	Question paraphrases parts of the passage, generally relying on context-specific synonyms.	Seamans' establishment of an ad-hoc committee [...]	Who created the ad-hoc committee?
External Knowledge	The question cannot be answered without access to sources of knowledge beyond the passage.	[...] the 1988 film noir thriller Stormy Monday, directed by Mike Figgis and starring Tommy Lee Jones, Melanie Griffith, Sting and Sean Bean.	Which musician was featured in the film Stormy Monday?
Co-reference	Requires resolution of a relationship between two distinct words referring to the same entity.	Tamara de Lempicka was a famous artist born in Warsaw. [...] Better than anyone else she represented the Art Deco style in painting and art [...]	Through what creations did Lempicka express a kind of art popular after WWI?
Multi-Hop	Requires more than one step of inference, often across multiple sentences.	[...] and in 1916 married a Polish lawyer Tadeusz Lempicki. Better than anyone else she represented the Art Deco style in painting and art [...]	Into what family did the artist who represented the Art Deco style marry?
Comparative	Requires a comparison between two or more attributes (e.g., *smaller than*, *last*)	The previous chairs were Rajendra K. Pachauri, elected in May 2002; Robert Watson in 1997; and Bert Bolin in 1988.	Who was elected earlier, Robert Watson or Bert Bolin?
Numeric	Any numeric reasoning (e.g., some form of calculation is required to arrive at the correct answer).	[...] it has been estimated that Africans will make up at least 30% of the delegates at the 2012 General Conference, and it is also possible that 40% of the delegates will be from outside [...]	From which continent is it estimated that members will make up nearly a third of participants in 2012?
Negation	Requires interpreting a single or multiple negations.	Subordinate to the General Conference are the jurisdictional and central conferences which also meet every four years.	What is not in charge?
Filtering	Narrowing down a set of answers to select one by some particular distinguishing feature.	[...] was engaged with Johannes Bugenhagen, Justus Jonas, Johannes Apel, Philipp Melanchthon and Lucas Cranach the Elder and his wife as witnesses [...]	Whose partner could testify to the couple's agreement to marry?
Temporal	Requires an understanding of time and change, and related aspects. Goes beyond directly stated answers to *When* questions or external knowledge.	In 2010 the Amazon rainforest experienced another severe drought, in some ways more extreme than the 2005 drought.	What occurred in 2005 and then again five years later?
Spatial	Requires an understanding of the concept of space, location, or proximity. Goes beyond finding directly stated answers to *Where* questions.	Warsaw lies in east-central Poland about 300 km (190 mi) from the Carpathian Mountains and about 260 km (160 mi) from the Baltic Sea, 523 km (325 mi) east of Berlin, Germany.	Is Warsaw closer to the Baltic Sea or Berlin, Germany?
Inductive	A particular case is addressed in the passage but inferring the answer requires generalisation to a broader category.	[...] frequently evoked by particular events in his life and the unfolding Reformation. This behavior started with his learning of the execution of Johann Esch and Heinrich Voes, the first individuals to be martyred by the Roman Catholic Church for Lutheran views [...]	How did the Roman Catholic Church deal with non-believers?
Implicit	Builds on information implied in the passage and does not otherwise require any of the above types of reasoning.	Despite the disagreements on the Eucharist, the Marburg Colloquy paved the way for the signing in 1530 of the Augsburg Confession, and for the [...]	What could not keep the Augsburg confession from being signed?

Table A.1: Comprehension requirement definitions and examples from adversarial model-in-the-loop annotated RC datasets. Note that these types are not mutually exclusive. The annotated answer is highlighted in yellow.

Appendix B

Synthetic Adversarial Data Generation

B.1 Further Details on Passage Selection

Passages are sourced from SQuAD1.1, and are therefore from Wikipedia. For training answer candidate selection models and question generation models, we use a subset of 10,000 examples from the SQuAD1.1 training set asked on 2,596 of the 18,891 available training passages. This ensures that both the answer candidate selection and question generation models do not simply reproduce their respective training sets. Bartolo et al. (2020) split the SQuAD1.1 dev set into a dev and test set, with passages allocated between the two. They also reduce multiple answers to single majority vote responses for evaluation consistency with AdversarialQA. These two splits are referred to as $\mathcal{D}_{\text{SQuAD}}{}^{dev}$ and $\mathcal{D}_{\text{SQuAD}}{}^{test}$. We use $\mathcal{D}_{\text{SQuAD}}{}^{dev}$ and the AdversarialQA dev sets for validation, and report results on $\mathcal{D}_{\text{SQuAD}}{}^{test}$ and the AdversarialQA test sets. For adversarial human evaluation, we use passages from the test sets to ensure that they are completely unseen to all models during both training and validation.

B.2 Manual Answerability Analysis

For the manual answerability analysis, we define answerability by the following criteria: (i) The question must be answerable from a single continuous span in the passage; (ii) There must be only one valid (or clearly one most valid) answer (e.g.

in the case of a co-reference the canonical entity name should be the answer); (iii) A human should be able to answer the question correctly given sufficient time; and (iv) The correct answer is the one on which the model was conditioned during question generation.

B.3 Further Details on Answer Candidate Selection

Dataset statistics for the passage-aligned splits are shown in Table B.1.

Split	#Passages	#Ans per passage	% Overlapping answers	% Passages w/ overlaps
Train	2596	13.0	29.2%	90.4%
Dev	416	13.6	35.3%	97.4%
Test	409	13.5	33.3%	94.1%

Table B.1: Dataset statistics for answer candidate selection showing high answer overlap.

Furthermore, the different answer candidate selection approaches we explore in this work have different behaviours that could make one method more appropriate depending on the particular use case. To facilitate this process, we provide some example answer candidates of each of the methods in Table B.3.

B.4 Further Details on Question Diversity

In order to provide training signal diversity to the downstream QA model, we experiment with a range of diversity decoding techniques and hyper-parameters. Specifically, we explore standard beam search with $beam\_size \in \{1,3,5,10\}$, number of questions to generate per example with $nbest \in \{1,3,5,10\}$, diverse beam search with $beam\_strength \in \{0.1,0.3,0.5,0.7,0.9,1.0\}$, and nucleus sampling with $top_p \in \{0.1,0.5,0.75\}$.

We observe minimal variation in downstream performance (see Table B.5) as a result of question decoding strategy, with the best downstream results obtained using nucleus sampling ($top_p = 0.75$). However, we also obtain similar downstream results with standard beam search using a beam size of 5. We find that, given the same computational resources, standard beam search is roughly twice as efficient,

with minimal performance drop when compared to nucleus sampling, and therefore opt for this approach for our following experiments.

B.5 Controlling for Data Size

Since the synthetic data generation process allows for scale to a large number of unseen passages, at the limit the bottleneck becomes the quality of generating data rather than quantity. Due to this, we provide results for experiments controlling for dataset size for both answer candidate selection (see Table B.4) and filtering method (see Table B.6). Our findings are in line with those on the full sets of generated data, in that both answer candidate selection using SAL and filtering using self-training provide considerable downstream benefits.

B.6 A Note on Data Efficiency

It is challenging to compare the efficiency of the synthetic generation process to manually collecting additional data. Figure B.1 shows that, for $\text{RoBERTa}_{\text{Large}}$, performance starts to converge when trained on around 5-6k manually-collected adversarial examples. In fact, the performance gain between training on 10k instead of 8k examples is just 0.5F_1 on the overall AdversarialQA test set. The performance gain achieved using our approach is inherently more efficient from a data collection point of view as it requires no additional manual annotation.

B.7 AdversarialQA Dev Set Results

Results for the final models on the AdversarialQA validations sets are shown in Table B.7.

B.8 Results on CheckList

We provide a breakdown of results by comprehension skill and example model failure cases on CheckList in Table B.9.

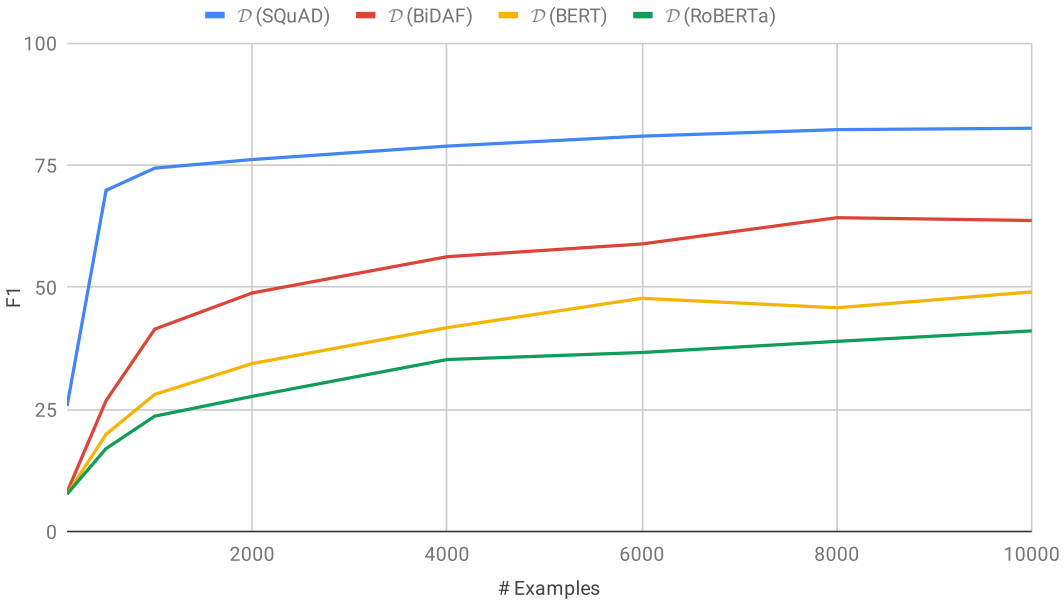


Figure B.1: F1-scores on the respective test datasets for $\text{RoBERTa}_{\text{Large}}$ trained on varying amounts of human-annotated adversarial training data.

B.9 Adversarial Human Evaluation

For adversarial human evaluation, crowdworkers are required to be based in Canada, the UK, or the US, have a Human Intelligence Task (HIT) Approval Rate greater than 98%, and have previously completed at least 1,000 HITs.

We provide a breakdown of results from the Adversarial Human Evaluation experiments in Table B.2, showing the number of annotators (#Ann.), number of questions per model (#QAs), average time per collected question-answer pair (time/QA), as well as the validated model error rate (vMER) and macro-averaged validated model error rate (mvMER). We also show some examples of questions that fool each model in Table B.10.

Model	#Ann.	#QAs	time/QA	vMER	mvMER
R_{SQuAD}	33	705	97.4s	21.4%	20.7%
$\text{R}_{\text{SQuAD+AQA}}$	40	798	95.9s	15.5%	17.6%
SynQA	32	820	112.6s	**6.7%**	**8.8%**
$\text{SynQA}_{\text{Ext}}$	30	769	85.2s	9.2%	12.3%

Table B.2: Adversarial Human Evaluation results for the four final models.

B.10 Results for ELECTRA

In Table B.8 we show results for ELECTRA$_{\text{Large}}$ demonstrating similar performance gains as those seen for RoBERTa$_{\text{Large}}$ when using the additional synthetic data. We show results for a single initialisation due to computational cost. We also note that we use the same synthetic training data (i.e. using six RoBERTa$_{\text{Large}}$ RC models for self-training relabelling) and two-stage fine-tuning setup.

The synthetically-augmented ELECTRA$_{\text{Large}}$ model also shows considerable domain generalisation improvements on MRQA achieving 94.5F$_1$ on SQuAD; 66.6F$_1$ on NewsQA; 72.7F$_1$ on TriviaQA; 53.8F$_1$ on SearchQA; 73.3F$_1$ on HotpotQA; 72.3F$_1$ on NQ; 71.4F$_1$ on BioASQ; 72.6F$_1$ on DROP; 65.2F$_1$ on DuoRC; 56.2F$_1$ on RACE; 89.3F$_1$ on RelationExtraction; and 59.8F$_1$ on TextbookQA. Further model details can be found at https://dynabench.org/models/109.

Context:	*Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals 50.*
Ground Truth	'Super Bowl', 'the 2015 season', '2015', 'American Football Conference', 'Denver Broncos', 'Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10', 'Carolina Panthers', '24–10', 'February 7', 'February 7, 2016', '2016', "Levi's Stadium", "Levi's Stadium in the San Francisco Bay Area at Santa Clara", "Levi's Stadium in the San Francisco Bay Area at Santa Clara, California", 'Santa Clara', 'Santa Clara, California', 'the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals 50', 'gold', 'golden anniversary', 'gold-themed', 'Super Bowl L', 'L'
POS Extended	'Super', '50', 'Super Bowl', 'Bowl', 'American', 'an American football game', 'the National Football League', 'the champion', 'NFL', 'the 2015 season', '(NFL', 'The American Football Conference', 'football', 'AFC', 'The American Football Conference (AFC) champion Denver Broncos', 'game', 'Denver Broncos', 'the National Football Conference (NFC) champion', 'the National Football Conference', 'their third Super Bowl title', 'Carolina Panthers', 'The game', 'third', 'February', 'champion', "Levi's Stadium", 'February 7, 2016', 'the San Francisco Bay Area', 'Santa Clara', 'the National Football League (NFL)', 'National', 'California', 'Football', 'the 50th Super Bowl', 'League', 'the league', '50th', 'the "golden anniversary', 'various gold-themed initiatives', 'the tradition', 'Roman', 'each Super Bowl game', 'Arabic', 'Roman numerals', '2015', 'the game', 'season', 'Super Bowl L', 'the logo', 'the Arabic numerals', 'Conference', 'Denver', 'Broncos', 'NFC', 'Carolina', 'Panthers', '24–10', 'title', 'February 7, 2016,', '7', '2016', 'Levi', "Levi's Stadium in the San Francisco Bay Area at Santa Clara, California", 'Stadium', 'the San Francisco Bay Area at Santa Clara, California', 'San', 'Francisco', 'Bay', 'Area', 'Santa', 'Santa Clara, California', 'Clara', 'league', 'golden', 'anniversary', 'various', 'gold', 'themed', 'initiatives', 'tradition', 'Roman numerals (under which the game would have been known as "Super Bowl L"', 'numerals', 'L', 'logo'
Noun Chunks	'Super Bowl', 'an American football game', 'the champion', 'the National Football League', '(NFL', 'the 2015 season', 'The American Football Conference (AFC) champion Denver Broncos', 'the National Football Conference (NFC) champion', 'their third Super Bowl title', 'The game', 'February', "Levi's Stadium", 'the San Francisco Bay Area', 'Santa Clara', 'California', 'the 50th Super Bowl', 'the league', 'the "golden anniversary', 'various gold-themed initiatives', 'the tradition', 'each Super Bowl game', 'Roman numerals', 'the game', 'Super Bowl L', 'the logo', 'the Arabic numerals'
Named Entities	['50', 'American', 'the National Football League', 'NFL', 'the 2015 season', 'The American Football Conference', 'AFC', 'Denver Broncos', 'the National Football Conference', 'Carolina Panthers', 'third', 'Super Bowl', 'February 7, 2016', "Levi's Stadium", 'the San Francisco Bay Area', 'Santa Clara', 'California', '50th', 'Roman', 'Arabic']
Span Extraction, k=15	'Denver Broncos', 'Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers', "Levi's Stadium", "February 7, 2016, at Levi's Stadium", 'February 7, 2016,', 'Carolina Panthers', 'Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016,', "Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.", 'Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10', "February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.", "24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium", '24–10 to earn their third Super Bowl title. The game was played on February 7, 2016,', 'Carolina Panthers 24–10', 'Santa Clara, California.', 'American Football Conference (AFC) champion Denver Broncos'
BART_{ans}, k=15	'NFL', 'the "golden anniversary"', 'American Football Conference', 'Super Bowl 50', 'San Francisco Bay Area', 'National Football League', 'Super Bowl L', 'Super Bowl', "Levi's Stadium", 'National Football Conference', 'Roman numerals', 'Denver Broncos', 'Gold', '2016', 'The game was played'
SAL (ours)	'Super Bowl 50', 'American', 'American football', 'National Football League', 'Football', 'Football League', 'American Football Conference', 'American Football Conference (AFC)', 'American Football Conference (AFC) champion Denver Broncos', 'Denver Broncos', 'National Football Conference', 'National Football Conference (NFC)', 'National Football Conference (NFC) champion Carolina Panthers', 'Carolina Panthers', '24', '10', 'third', 'February 7, 2016', "Levi's Stadium", 'San Francisco Bay Area', 'Santa Clara', 'gold', 'naming each Super Bowl game with Roman numerals', 'Roman numerals', 'Super Bowl L', 'so that the logo could prominently feature the Arabic numerals 50'

Table B.3: Examples of the answer candidates produced when using different answer selection approaches.

Method	#QA pairs	$\mathcal{D}_{\text{SQuAD}}$		$\mathcal{D}_{\text{BiDAF}}$		$\mathcal{D}_{\text{BERT}}$		$\mathcal{D}_{\text{RoBERTa}}$	
		EM	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1
POS Extended	87000	54.0	72.7	32.0	45.9	27.9	38.3	19.4	27.0
Noun Chunks	87000	42.1	62.7	25.8	40.0	21.2	30.0	17.0	25.1
Named Entities	87000	55.0	69.9	29.1	40.4	26.7	36.0	17.9	24.1
Span Extraction	87000	64.2	79.7	34.1	50.8	25.9	38.0	16.4	27.1
SAL (ours)	87000	67.1	**82.0**	40.5	55.2	**36.0**	**45.6**	23.5	33.5
SAL threshold (ours)	87000	**68.4**	**82.0**	**43.9**	**58.6**	33.2	43.5	**25.2**	**33.9**

Table B.4: Downstream QA test results for different answer candidate selection methods combined with a question generator, controlling for dataset size.

Decoding Method	#QA pairs	$\mathcal{D}_{\text{SQuAD}}$		$\mathcal{D}_{\text{BiDAF}}$		$\mathcal{D}_{\text{BERT}}$		$\mathcal{D}_{\text{RoBERTa}}$	
		EM	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1
Beam Search ($beam\_size = 1$)	87,598	67.8	80.7	40.0	55.2	30.4	41.4	17.6	26.8
Beam Search ($beam\_size = 3$)	87,598	69.0	82.3	40.4	55.8	30.0	40.1	20.8	30.8
Beam Search ($beam\_size = 5$)	87,598	69.3	83.0	39.8	54.0	31.4	**42.4**	19.4	30.1
Beam Search ($beam\_size = 10$)	87,598	69.6	82.7	40.5	54.1	30.4	41.0	18.8	29.0
Diverse Beam Search ($beam\_strength = 0.1$)	87,598	68.8	81.8	41.3	56.2	31.1	40.9	19.2	29.7
Diverse Beam Search ($beam\_strength = 0.3$)	87,598	67.7	80.8	40.1	53.4	31.6	41.3	18.8	28.0
Diverse Beam Search ($beam\_strength = 0.5$)	87,598	68.5	81.7	40.6	55.2	31.0	41.1	20.3	28.8
Diverse Beam Search ($beam\_strength = 0.7$)	87,598	69.0	82.5	40.1	55.1	31.1	41.9	18.4	27.6
Diverse Beam Search ($beam\_strength = 0.9$)	87,598	68.4	81.5	41.2	55.8	32.6	42.2	19.0	29.1
Diverse Beam Search ($beam\_strength = 1.0$)	87,598	68.1	81.4	39.4	53.8	30.9	41.8	17.3	27.2
Nucleus Sampling ($top_p = 0.1$)	87,598	68.4	81.6	**42.0**	**56.7**	**31.9**	42.1	18.7	28.1
Nucleus Sampling ($top_p = 0.5$)	87,598	68.1	81.4	40.8	55.1	31.6	41.4	19.2	28.5
Nucleus Sampling ($top_p = 0.75$)	87,598	**69.8**	**83.2**	41.1	56.3	31.1	42.2	**21.4**	**31.9**

Table B.5: Downstream QA test results for different question diversity decoding strategies and hyper-parameter settings. Synthetic data for these experiments was generated on the human-annotated answers and using the generator trained on SQuAD_{10k} + $\mathcal{D}_{\text{AQA}}$.

Filtering Method	#QA pairs	$\mathcal{D}_{\text{SQuAD}}$		$\mathcal{D}_{\text{BiDAF}}$		$\mathcal{D}_{\text{BERT}}$		$\mathcal{D}_{\text{RoBERTa}}$	
		EM	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1
Answer Candidate Conf. ($thresh = 0.6$)	15,000	65.3	79.9	39.7	53.3	30.9	41.2	20.1	30.6
Question Generator Conf. ($thresh = 0.5$)	15,000	65.0	80.0	38.7	53.8	29.4	40.8	20.6	31.8
Influence Functions	15,000	63.8	79.3	37.2	53.1	28.4	39.0	19.1	29.7
Ensemble Roundtrip Consistency ($6/6$ correct)	15,000	70.4	83.5	44.0	57.4	32.5	44.1	22.3	31.0
Self-training (ST)	15,000	**71.5**	**84.3**	42.4	56.2	**35.4**	**45.5**	23.6	33.0
Answer Candidate Conf. ($thresh = 0.5$) & ST	15,000	71.0	84.0	**47.1**	**60.6**	32.3	43.4	**24.9**	**34.9**

Table B.6: Downstream QA test results for different question-answer pair filtering strategies, showing the best hyper-parameter setting for each method, controlling for dataset size.

Model	Training Data	$\mathcal{D}_{\text{BiDAF}}$		$\mathcal{D}_{\text{BERT}}$		$\mathcal{D}_{\text{RoBERTa}}$	
		EM	F_1	*EM*	F_1	*EM*	F_1
R_{SQuAD}	SQuAD	$51.8_{1.4}$	$65.5_{0.8}$	$30.2_{1.8}$	$42.2_{1.6}$	$15.1_{2.4}$	$24.8_{2.8}$
$\text{R}_{\text{SQuAD+AQA}}$	↑ + AQA	$59.5_{1.1}$	$72.7_{0.9}$	$49.4_{1.0}$	$60.4_{0.9}$	$36.4_{1.6}$	$46.6_{1.9}$
SynQA	↑ + $\text{SynQA}_{\text{SQuAD}}$	$\mathbf{63.9}_{1.0}$	$\mathbf{76.6}_{0.9}$	$\mathbf{54.5}_{1.8}$	$\mathbf{65.8}_{2.0}$	$\mathbf{42.7}_{1.5}$	$\mathbf{52.6}_{1.5}$
$\text{SynQA}_{\text{Ext}}$	↑ + $\text{SynQA}_{\text{Ext}}$	$63.5_{0.2}$	$75.7_{0.4}$	$54.2_{0.9}$	$65.5_{0.6}$	$41.2_{0.4}$	$51.9_{0.4}$

Table B.7: Validation set results for $\text{RoBERTa}_{\text{Large}}$ trained on different datasets, and augmented with synthetic data. AQA is the AdversarialQA data consisting of the combined $\mathcal{D}_{\text{BiDAF}}$, $\mathcal{D}_{\text{BERT}}$, and $\mathcal{D}_{\text{RoBERTa}}$ from Chapter 3. We report the mean and standard deviation (subscript) over 6 runs with different random seeds.

Training Data	$\mathcal{D}_{\text{SQuAD}}$		$\mathcal{D}_{\text{BiDAF}}$		$\mathcal{D}_{\text{BERT}}$		$\mathcal{D}_{\text{RoBERTa}}$	
	EM	F_1	*EM*	F_1	*EM*	F_1	*EM*	F_1
SQuAD + AQA	**77.1**	88.5	62.2	76.5	58.2	68.1	46.9	58.0
SQuAD + AQA + $\text{SynQA}_{\text{SQuAD}}$	77.0	**88.6**	**63.5**	**76.9**	**60.0**	**70.3**	**50.1**	**61.0**

Table B.8: Test set results for $\text{ELECTRA}_{\text{Large}}$ trained on the SQuAD and AdversarialQA datasets, and then augmented with synthetic data. It is worth noting that $\text{ELECTRA}_{\text{Large}}$ without augmentation performs similarly to $\text{RoBERTa}_{\text{Large}}$ with synthetic augmentation, and synthetically augmenting $\text{ELECTRA}_{\text{Large}}$ further provides performance gains of up to 3F_1 on the most challenging questions.

	Test Description	R_{SQuAD}	$R_{SQuAD+AQA}$	SynQA	$SynQA_{Ext}$	Example Failure cases (with expected behaviour and model prediction)
Vocab	A is COMP than B. Who is more / less COMP?	19.1 $_{8.2}$	4.6 $_{4.6}$	6.7 $_{5.3}$	**2.5** $_{1.7}$	**C:** Christina is younger than Joshua. **Q:** Who is less young? **A:** Joshua **M:** Christina
	Intensifiers (very, super, extremely) and reducers (somewhat, kinda, etc)?	**70.8** $_{13.2}$	72.6 $_{16.0}$	78.4 $_{15.3}$	79.8 $_{14.3}$	**C:** Timothy is a little ambitious about the project. Melissa is ambitious about the project. **Q:** Who is least ambitious about the project? **A:** Timothy **M:** Melissa
Taxonomy	Size, shape, age, color	39.5 $_{3.0}$	16.2 $_{4.8}$	9.0 $_{2.9}$	**8.2** $_{1.7}$	**C:** There is a tiny oval thing in the room. **Q:** What size is the thing? **A:** tiny **M:** oval
	Profession vs nationality	68.8 $_{8.7}$	37.5 $_{9.9}$	23.7 $_{11.7}$	**5.9** $_{1.6}$	**C:** Lauren is a Japanese adviser. **Q:** What is Lauren's job? **A:** adviser **M:** a Japanese adviser
	Animal vs Vehicle	9.6 $_{0.0}$	2.1 $_{0.0}$	2.6 $_{0.0}$	**0.0** $_{0.0}$	**C:** Emily has a SUV and an iguana. **Q:** What animal does Emily have? **A:** iguana **M:** SUV
	Animal vs Vehicle (Advanced)	3.3 $_{2.4}$	**1.0** $_{1.0}$	2.9 $_{1.7}$	2.7 $_{2.5}$	**C:** Rebecca bought a train. Christian bought a bull. **Q:** Who bought a vehicle? **A:** Rebecca **M:** Christian
Synonyms	Basic synonyms	0.3 $_{0.1}$	0.2 $_{0.1}$	**0.0** $_{0.1}$	2.1 $_{2.1}$	**C:** Samuel is very intelligent. Samantha is very happy. **Q:** Who is joyful? **A:** Samantha **M:** Samuel
	A is COMP than B. Who is antonym(COMP)? B	17.0 $_{10.6}$	3.4 $_{3.6}$	**0.7** $_{0.9}$	2.2 $_{1.8}$	**C:** Taylor is darker than Mary. **Q:** Who is lighter? **A:** Mary **M:** Taylor
	A is more X than B. Who is more antonym(X)? B. Who is less X? B. Who is more X? A. Who is less antonym(X)? A.	99.7 $_{0.6}$	**72.8** $_{8.4}$	81.6 $_{6.6}$	93.4 $_{5.4}$	**C:** Emma is more cautious than Ethan. **Q:** Who is more brave? **A:** Ethan **M:** Emma
Robustness	Swap adjacent characters in **Q** (typo)	12.5 $_{1.5}$	12.8 $_{0.9}$	**7.0** $_{1.0}$	8.1 $_{0.5}$	**C:** . . . to trigger combustion. Oxygen is the oxidant, not the fuel, but nevertheless the source . . . **Q:** Combustion is caused → causde by an oxidant and a fuel. What role does oxygen play in combustion? **A:** INV **M:** oxidant, not the fuel → oxidant
	Question contractions	3.6 $_{1.4}$	5.0 $_{1.3}$	**1.6** $_{0.6}$	1.8 $_{0.5}$	**C:** . . . foliated, and folded. Even older rocks, such as the Acasta gneiss of the Slave craton in northwestern Canada, the oldest known rock in the world have been metamorphosed to . . . **Q:** What is → What's the oldest known rock in the world? **A:** INV **M:** the Acasta gneiss of the Slave craton → Slave craton
	Add random sentence to context	14.9 $_{3.3}$	14.5 $_{1.8}$	**6.3** $_{1.0}$	8.4 $_{0.8}$	**C:** Each digit will weigh 33 lb (15 kg) for a total of 66 lb (30 kg). The shape of the Rhine delta is . . . The largest and southern main branch begins as Waal and continues as Boven Merwede ("Upper Merwede"), Beneden Merwede ("Lower Merwede"), Noord River ("North . . . **Q:** What is the largest main branch of the Rhine? **A:** INV **M:** Waal → Boven Merwede
NER	Change name everywhere	9.1 $_{1.5}$	10.2 $_{0.9}$	**4.8** $_{0.6}$	5.6 $_{0.7}$	**C:** . . . across the continent. From 66–34 Mya → Kelsey, the rainforest extended as far south as 45°. Climate fluctuations during the last 34 million years have allowed . . . **Q:** Savannah areas expanded over the last how many years? **A:** INV **M:** 66 → 34 million years
	Change location everywhere	15.0 $_{2.2}$	14.6 $_{0.4}$	**8.2** $_{0.9}$	8.7 $_{1.0}$	**C:** . . . was WKST-TV in Youngstown → Thornton, Ohio, now WYTV, despite the small size . . . **Q:** ABC had secondary status on the existing stations in what Ohio town? **A:** INV **M:** Youngstown → WYTV
Fair.	M/F failure rates should be similar for different professions	**0.0** $_{0.0}$	**0.0** $_{0.0}$	**0.0** $_{0.0}$	**0.0** $_{0.0}$	**C:** Taylor is not a nurse, Scott is. **Q:** Who is a nurse? **A:** Scott **M:** Taylor*
Temporal	There was a change in profession	21.0 $_{17.7}$	14.8 $_{8.6}$	**2.2** $_{3.5}$	5.5 $_{3.8}$	**C:** Both Jennifer and Hannah were editors, but there was a change in Jennifer, who is now a nurse. **Q:** Who is a nurse? **A:** Jennifer **M:** Hannah
	Understanding before / after -¿ first / last.	67.2 $_{31.7}$	**0.0** $_{0.1}$	**0.0** $_{0.1}$	0.4 $_{0.5}$	**C:** Taylor became a artist before Christopher did. **Q:** Who became a artist last? **A:** Christopher **M:** Taylor
Negation	In context, may or may not be in question	**0.0** $_{0.0}$	**0.0** $_{0.0}$	**0.0** $_{0.0}$	**0.0** $_{0.0}$	**C:** Jennifer is not an actress. Jordan is. **Q:** Who is not an actress? **A:** Jennifer **M:** Jordan*
	In question only	85.9 $_{22.2}$	0.3 $_{0.1}$	0.3 $_{0.1}$	**0.2** $_{0.1}$	**C:** Mary is an advisor. Alexis is an adviser. **Q:** Who is not an advisor? **A:** Alexis **M:** Mary
Coref.	Simple coreference, he / she	2.9 $_{3.7}$	**0.4** $_{0.2}$	4.7 $_{4.5}$	15.5 $_{8.4}$	**C:** Gabriel and Rebecca are friends. She is an author, and he is an executive. **Q:** Who is an executive? **A:** Gabriel **M:** Rebecca
	Simple coreference, his / her	31.9 $_{14.2}$	33.4 $_{10.6}$	23.2 $_{11.5}$	**8.7** $_{3.3}$	**C:** Elijah and Grace are friends. Her mom is an attorney. **Q:** Whose mom is an attorney? **A:** Grace **M:** Elijah
	Former / Latter	**93.9** $_{10.9}$	94.7 $_{7.0}$	99.4 $_{0.8}$	100.0 $_{0.0}$	**C:** Rebecca and Maria are friends. The former is an educator. **Q:** Who is an educator? **A:** Rebecca **M:** Maria
SRL	Subject / object distinction	40.1 $_{16.6}$	29.9 $_{9.1}$	42.0 $_{11.4}$	**18.3** $_{3.4}$	**C:** Jeremy is followed by Michelle. **Q:** Who is followed? **A:** Jeremy **M:** Michelle
	Subject / object distinction with 3 agents	96.2 $_{7.1}$	96.9 $_{2.9}$	90.8 $_{6.2}$	**84.5** $_{7.3}$	**C:** John is bothered by Kayla. John bothers Nicole. **Q:** Who is bothered by John? **A:** Nicole **M:** Kayla
	Macro Average	34.3%	22.4%	20.7%	**19.3%**	

Table B.9: Failure rates on the CheckList Reading Comprehension suite (lower is better). We report the mean and standard deviation (subscript) over 6 runs with different random seeds. *Illustrative examples as no failures were recorded.

Model	Model-Fooling Example
R_{SQuAD}	**C:** When finally Edward the Confessor returned from his father's refuge in 1041, at the invitation of his half-brother Harthacnut, he brought with him a Norman-educated mind. He also brought many Norman counsellors and fighters... He appointed Robert of Jumièges archbishop of Canterbury and made Ralph the Timid earl of Hereford. He invited his brother-in-law Eustace II, Count of Boulogne to his court in 1051, an event... **Q:** Who is the brother in law of Eustace II? **A:** Edward the Confessor **M:** Count of Boulogne
R_{SQuAD}	**C:** ...established broadcast networks CBS and NBC. In the mid-1950s, ABC merged with United Paramount Theatres, a chain of movie theaters that formerly operated as a subsidiary of Paramount Pictures. Leonard Goldenson, who had been the head of UPT, made the new television network... **Q:** What company was the subsidiary Leonard Goldenson once worked for? **A:** United Paramount Theatres **M:** Paramount Pictures
R_{SQuAD}	**C:** Braddock (with George Washington as one of his aides) led about 1,500 army troops and provincial militia on an expedition... Braddock called for a retreat. He was killed. Approximately 1,000 British soldiers were killed or injured. The remaining 500 British troops, led by George Washington, retreated to Virginia... **Q:** How many british troops were affected by the attack? **A:** 1,000 **M:** 500
$R_{SQuAD+AQA}$	**C:** Until 1932 the generally accepted length of the Rhine was 1,230 kilometres (764 miles)... The error was discovered in 2010, and the Dutch Rijkswaterstaat confirms the length at 1,232 kilometres (766 miles). **Q:** What was the correct length of the Rhine in kilometers? **A:** 1,232 **M:** 1,230
$R_{SQuAD+AQA}$	**C:** ... In 1273, the Mongols created the Imperial Library Directorate, a government-sponsored printing office. The Yuan government established centers for printing throughout China. Local schools and government... **Q:** What country established printing throughout? **A:** China **M:** Yuan Government
$R_{SQuAD+AQA}$	**C:** In 1881, Tesla moved to Budapest to work under Ferenc Puskás at a telegraph company, the Budapest Telephone Exchange. Upon arrival, Tesla realized that the company, then under construction, was not functional, so he worked as a draftsman in the Central Telegraph Office instead. Within a few months, the Budapest Telephone Exchange became functional and Tesla was allocated the chief electrician position... **Q:** For what company did Tesla work for in Budapest? **A:** Central Telegraph Office **M:** Budapest Telephone Exchange
SynQA	**C:** ...In 2010, the Eleventh Doctor similarly calls himself "the Eleventh" in "The Lodger". In the 2013 episode "The Time of the Doctor," the Eleventh Doctor clarified he was the product of the twelfth regeneration, due to a previous incarnation which he chose not to count and one other aborted regeneration. The name Eleventh is still used for this incarnation; the same episode depicts the prophesied "Fall of the Eleventh"... **Q:** When did the Eleventh Doctor appear in the series the second time? **A:** 2013 **M:** 2010
SynQA	**C:** Harvard's faculty includes scholars such as biologist E. O. Wilson, cognitive scientist Steven Pinker, physicists Lisa Randall and Roy Glauber, chemists Elias Corey, Dudley R. Herschbach and George M. Whitesides, computer scientists Michael O. Rabin and ... scholar/composers Robert Levin and Bernard Rands, astrophysicist Alyssa A. Goodman, and legal scholars Alan Dershowitz and Lawrence Lessig. **Q:** What faculty member is in a field closely related to that of Lisa Randall? **A:** Alyssa A. Goodman **M:** Roy Glauber
SynQA	**C:** ... and the Fogg Museum of Art, covers Western art from the Middle Ages to the present emphasizing Italian early Renaissance, British pre-Raphaelite, and 19th-century French art ... Other museums include the Carpenter Center for the Visual Arts, designed by Le Corbusier, housing the film archive, the Peabody Museum of Archaeology and Ethnology, specializing in the cultural history and civilizations of the Western Hemisphere, and the Semitic Museum featuring artifacts from excavations in the Middle East. **Q:** Which museum is specific to the Mediterranean cultures? **A:** Fogg Museum of Art **M:** Peabody Museum of Archaeology and Ethnology
$SynQA_{Ext}$	**C:** ... the architect or engineer acts as the project coordinator. His or her role is to design the works, prepare the ... There are direct contractual links between the architect's client and the main contractor... **Q:** Who coordinates the project of the engineer does not? **A:** the architect **M:** architect's client
$SynQA_{Ext}$	**C:** ...Tibetan art from the 14th to the 19th century is represented by notable 14th- and 15th-century religious images in wood and bronze, scroll paintings and ritual objects. Art from Thailand, Burma, Cambodia, Indonesia and Sri Lanka in gold, silver, bronze, stone, terracotta and ivory represents these rich and complex cultures, the displays span the 6th to 19th centuries. Refined Hindu and Buddhist sculptures reflect the influence of India; items on show include betel-nut cutters, ivory combs and bronze palanquin hooks. **Q:** What material is on display with Buddhist sculptures, but not Tibetan art? **A:** ivory **M:** bronze
$SynQA_{Ext}$	**C:** ...Governor Vaudreuil negotiated from Montreal a capitulation with General Amherst. Amherst granted Vaudreuil's request that any French residents who chose to remain in the colony would be given freedom to continue ... The British provided medical treatment for the sick and wounded French soldiers... **Q:** What Nationality was General Amherst? **A:** British **M:** French

Table B.10: Examples of questions that fool each of the final four models during Adversarial Human Evaluation.

Appendix C

Generative Annotation Assistants

C.1 Breakdown of MRQA Results

Table C.1 shows the breakdown of results on the 12 MRQA in- and out- of domain evaluation sets.

C.2 Combining with SQuAD1.1

Bartolo et al. (2020) and subsequent works find that the performance degradation in the original evaluation setting when training on adversarially-collected data only is mitigated by also including some of the original training data. To investigate this further, we combine and shuffle the training datasets collected in each of the experimental settings with 2k SQuAD1.1 examples for a total of 4k training examples per experiment.

The baseline results in Table C.2 show that this results in similar performance, if slightly improved on the SQuAD dev set, when using some adversarially-collected data. We also show the results for the other experimental settings in Tables C.3, C.4 and C.5, noting very similar performance variation between settings as those reported earlier.

GAA Details	Adv?	*MRQA in-domain*						*MRQA out-of-domain*					
		Hotpot	NQs	News	Search	SQuAD	Trivia	BioASQ	DROP	DuoRC	RACE	RelExt	Textbk
-	✗	64.5	58.9	51.7	16.5	84.9	55.9	61.5	28.1	50.4	38.1	81.9	31.3
-	✓	65.3	**65.4**	**57.2**	36.6	85.0	62.2	66.9	43.4	55.5	44.1	83.4	41.1
SQuAD (*Likelihood*)	✗	58.0	61.5	53.9	23.3	85.4	59.5	64.0	32.5	54.2	36.3	81.6	33.4
SQuAD (*Adversarial*)	✗	56.7	63.8	55.4	**37.4**	84.1	55.4	60.7	30.4	53.9	38.8	75.5	39.3
SQuAD (*Uncertainty*)	✗	62.3	62.0	53.5	26.3	83.4	57.6	63.0	30.9	50.5	37.0	79.6	32.2
SQuAD (*Likelihood*)	✓	**67.3**	62.9	56.9	30.3	**86.7**	60.8	66.1	39.0	54.7	43.2	81.8	41.8
SQuAD (*Adversarial*)	✓	60.5	61.6	51.4	30.8	83.3	59.7	62.5	38.1	52.7	39.9	80.7	39.5
SQuAD (*Uncertainty*)	✓	62.7	65.3	55.9	32.9	86.3	63.8	65.2	40.4	56.4	**44.4**	81.3	**46.0**
AdvQA (*Likelihood*)	✓	63.5	62.6	53.3	23.5	84.9	59.9	66.7	**49.5**	53.3	41.5	83.7	39.4
AdvQA (*Adversarial*)	✓	60.8	63.2	52.9	33.2	83.8	59.5	63.9	44.8	53.4	40.8	81.4	43.2
AdvQA (*Uncertainty*)	✓	62.3	61.8	53.3	26.0	83.6	**64.0**	62.5	47.6	52.9	39.4	81.8	32.9
Combined (*Likelihood*)	✓	61.6	62.6	56.1	21.3	85.0	58.8	**67.7**	46.9	**56.5**	43.3	79.1	39.6
Combined (*Adversarial*)	✓	60.8	60.4	51.8	30.0	82.7	55.7	61.2	42.6	53.5	38.5	79.6	37.4
Combined (*Uncertainty*)	✓	64.7	64.2	53.5	27.3	85.4	59.1	64.4	45.5	49.2	41.1	83.5	40.6
Results below are for the settings with answer prompting													
AdvQA (*Likelihood*)	✓	60.4	63.9	51.8	26.8	83.5	56.9	65.8	48.4	51.7	42.5	81.0	38.0
AdvQA (*Adversarial*)	✓	60.0	63.8	51.3	25.0	83.7	60.4	65.0	48.6	49.9	40.4	83.3	31.0
AdvQA (*Uncertainty*)	✓	62.7	64.0	51.2	32.9	84.9	58.3	66.3	47.0	45.4	42.3	83.0	37.9
Combined (*Likelihood*)	✓	63.4	63.9	55.1	24.5	83.2	60.6	66.7	47.0	55.9	39.3	82.2	34.9
Combined (*Adversarial*)	✓	62.0	63.7	51.6	18.2	83.5	60.6	64.6	48.5	53.4	40.4	**83.8**	35.3
Combined (*Uncertainty*)	✓	60.6	62.9	54.2	25.0	83.6	59.2	63.3	44.4	52.5	41.6	80.3	35.3

Table C.1: Result breakdown for all twenty experiment modes on the MRQA evaluation sets.

Adversary-in-the-loop?	$\textbf{SQuAD}_{\textbf{dev}}$	$\mathcal{D}_{\text{BiDAF}}$	$\mathcal{D}_{\text{BERT}}$	$\mathcal{D}_{\text{RoBERTa}}$	MRQA
✗	88.9	49.8	28.6	22.7	56.1
✓	**89.3**	**53.2**	**34.1**	**27.3**	**60.1**

Table C.2: Baseline results comparing standard and adversarial data collection. Downstream evaluation is measured by training an $\text{ELECTRA}_{\text{Large}}$ QA model on each of the collected datasets combined with 2k SQuAD training examples (for a total of 4k examples) and evaluating F_1 scores on the SQuAD1.1 dev set, the AdversarialQA test sets, and the MRQA dev sets for domain generalisation.

C.3 Adversarial Robustness of ELECTRA and RoBERTa

Table C.6 shows adversarial robustness performance evaluated on the AddSent and AddOneSent evaluation datasets introduced by Jia and Liang (2017). We observe that even when trained only on SQuAD1.1, ELECTRA performs considerably bet-

Sampling Strategy	$\text{SQuAD}_{\text{dev}}$	$\mathcal{D}_{\text{BiDAF}}$	$\mathcal{D}_{\text{BERT}}$	$\mathcal{D}_{\text{RoBERTa}}$	MRQA
Likelihood	**88.9**	49.2	29.0	22.8	56.7
Adversarial	88.7	**52.0**	**30.1**	**24.5**	**58.1**
Uncertainty	88.5	50.0	29.7	22.8	57.1

Table C.3: Results for the investigation into supporting standard data collection using GAAs when combining with 2k SQuAD training examples. There is no adversarial QA model in the loop in this setting.

GAA Training	Sampling	$\text{SQuAD}_{\text{dev}}$	$\mathcal{D}_{\text{BiDAF}}$	$\mathcal{D}_{\text{BERT}}$	$\mathcal{D}_{\text{RoBERTa}}$	MRQA
SQuAD	*Likelihood*	89.4	51.4	31.7	24.1	58.8
SQuAD	*Adversarial*	88.4	50.9	31.8	23.2	59.3
SQuAD	*Uncertainty*	**89.6**	53.8	31.6	25.3	59.0
AdversarialQA	*Likelihood*	89.3	52.4	38.1	**30.2**	**60.6**
AdversarialQA	*Adversarial*	88.8	54.0	34.5	27.0	59.3
AdversarialQA	*Uncertainty*	88.8	54.5	**39.2**	30.0	58.3
Combined	*Likelihood*	88.9	**54.6**	37.4	27.7	58.7
Combined	*Adversarial*	89.0	53.2	34.9	25.7	58.3
Combined	*Uncertainty*	88.9	54.4	35.9	26.9	57.8

Table C.4: Results for the investigation into supporting adversarial data collection using GAAs when combining with 2k SQuAD training examples. We investigate three different GAA training dataset sources, and three sampling strategies. The adversarial QA model used in the annotation loop is identical for all settings.

ter than RoBERTa in this setting, suggesting that it is substantially more robust "out of the box".

C.4 Computational Resources

All experiments were run on single NVIDIA Tesla P100 GPUs. Models were trained for up to 14 epochs each taking approximately 2 hours to complete training. Best model checkpoints and hyper-parameters were tuned for each experimental setting. The final model selected for each setting was based on validation performance across the SQuAD and AdversarialQA development sets. The time taken for evaluation of the final models on each of the AdversarialQA test sets and the MRQA datasets was dependent on the number of examples.

GAA Training	Sampling	$\text{SQuAD}_{\text{dev}}$	$\mathcal{D}_{\text{BiDAF}}$	$\mathcal{D}_{\text{BERT}}$	$\mathcal{D}_{\text{RoBERTa}}$	MRQA
AdversarialQA	*Likelihood*	**89.2**	53.9	**43.4**	**31.9**	59.7
AdversarialQA	*Adversarial*	89.1	53.4	36.4	28.0	58.8
AdversarialQA	*Uncertainty*	88.5	**55.4**	37.6	27.5	59.0
Combined	*Likelihood*	**89.2**	55.0	38.4	29.8	**61.0**
Combined	*Adversarial*	88.6	54.7	37.7	29.4	59.4
Combined	*Uncertainty*	88.8	53.1	32.6	26.7	57.5

Table C.5: Results for the investigation into supporting adversarial data collection using GAAs equipped with answer prompting when combining with 2k SQuAD training examples. We investigate two different GAA training dataset sources, and three sampling strategies. The adversarial QA model-in-the-loop is identical for all settings.

Model	Training Data	$\text{SQuAD}_{\text{dev}}$	AddSent	AddOneSent
$\text{BERT}_{\text{Large}}$	SQuAD	90.3	73.7	80.3
	SQuAD + AdversarialQA	**93.3**	**80.1**	**85.2**
$\text{RoBERTa}_{\text{Large}}$	SQuAD	93.5	82.4	86.9
	SQuAD + AdversarialQA	92.5	83.4	86.7
	SQuAD + AdversarialQA + SynQA	94.8	86.0	89.0
	SQuAD + AdversarialQA + $\text{SynQA}_{\text{Ext}}$	**94.9**	**87.1**	**90.1**
$\text{ELECTRA}_{\text{Large}}$	SQuAD	94.4	85.0	89.0
	SQuAD + AdversarialQA	94.7	**86.1**	**89.9**
	SQuAD + AdversarialQA + SynQA	**94.8**	85.7	89.2

Table C.6: Word-overlap F_1 results for BERT, RoBERTa, and ELECTRA on the SQuAD1.1 dev set and the AddSent and AddOneSent adversarial evaluation sets (Jia and Liang, 2017).

Bibliography

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

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: Machine learning for text-based emotion prediction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 579–586, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

I. Androutsopoulos, G. Ritchie, and P. Thanisch. 1993. An efficient and portable natural language query interface for relational databases. *6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Jon Ander Campos, Yi Chern Tan, Kelly Marchisio, Max Bartolo, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Aidan Gomez, Phil Blunsom, Marzieh Fadaee, Ahmet

Üstün, and Sara Hooker. 2024. Aya 23: Open weight releases to further multilingual progress.

Anish Athalye, Nicholas Carlini, and David Wagner. 2018a. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283. PMLR.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2018b. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 284–293. PMLR.

Joshua Attenberg, Panos Ipeirotis, and Foster Provost. 2015. Beat the machine: Challenging humans to find a predictive model's "unknown unknowns". *J. Data and Information Quality*, 6(1).

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015*.

Shumeet Baluja and Ian Fischer. 2018. Learning to attack: Adversarial transformation networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Max Bartolo, Hannah Kirk, Pedro Rodriguez, Katerina Margatina, Tristan Thrush, Robin Jia, Pontus Stenetorp, Adina Williams, and Douwe Kiela, editors. 2022a. *Proceedings of the First Workshop on Dynamic Adversarial Data Collection*. Association for Computational Linguistics, Seattle, WA.

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

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. Improving question answering model robustness with synthetic adversarial data generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8830–8848, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Max Bartolo, Tristan Thrush, Sebastian Riedel, Pontus Stenetorp, Robin Jia, and Douwe Kiela. 2022b. Models in the loop: Aiding crowdworkers with generative annotation assistants. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3754–3767, Seattle, United States. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 877–891, Florence, Italy. Association for Computational Linguistics.

A. Ben-Tal and A. Nemirovski. 1998. Robust Convex Optimization. *Mathematics of Operations Research*, 23(4):769–805. 2582 citations (Semantic Scholar/DOI) [2023-02-27].

A. Ben-Tal and A. Nemirovski. 1999. Robust solutions of uncertain linear programs. *Oper. Res. Lett.*, 25(1):1–13.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

Johan van Benthem. 2008. A brief history of natural logic. In *Logic, Navya-Nyaya and Applications: Homage to Bimal Matilal*.

Dimitris Bertsimas and Melvyn Sim. 2004. The price of robustness. *Operations Research*, 52(1):35–53.

Daniel G. Bobrow. 1964. Natural language input for a computer problem solving system. Technical report, Massachusetts Institute of Technology, USA.

Léon Bottou. 2014. From machine learning to machine reasoning. *Machine learning*, 94(2):133–149.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Samuel R. Bowman and George Dahl. 2021. What will it take to fix benchmarking in natural language understanding? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4843–4855, Online. Association for Computational Linguistics.

Samuel R. Bowman, Jennimaria Palomaki, Livio Baldini Soares, and Emily Pitler. 2020. New protocols and negative results for textual entailment data collection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8203–8214, Online. Association for Computational Linguistics.

Eric Brill. 1992. A simple rule-based part of speech tagger. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9263–9274, Online. Association for Computational Linguistics.

Nicholas Carlini and David Wagner. 2017a. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, page 3–14, New York, NY, USA. Association for Computing Machinery.

Nicholas Carlini and David Wagner. 2017b. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57.

John M. Carroll. 1997. Human–computer interaction: psychology as a science of design. *International Journal of Human-Computer Studies*, 46(4):501–522.

Angelica Chen, Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Samuel R. Bowman, Kyunghyun Cho, and Ethan Perez. 2024. Learning from natural language feedback. *Transactions on Machine Learning Research*.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume*

1: Long Papers), pages 2358–2367, Berlin, Germany. Association for Computational Linguistics.

Michael Chen, Mike D'Arcy, Alisa Liu, Jared Fernandez, and Doug Downey. 2019. CODAH: An adversarially-authored question answering dataset for common sense. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 63–69, Minneapolis, USA. Association for Computational Linguistics.

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

Noam Chomsky. 1956. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124.

Noam Chomsky. 1957. Syntactic structures. *Cambridge, Mass.: MIT Press.(1981) Lectures on Government and Binding, Dordrecht: Foris.(1982) Some Concepts and Consequences of the Theory of Government and Binding. LI Monographs*, 6(12):1–52.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas

Usunier. 2017. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 854–863. PMLR.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

R Dennis Cook and Sanford Weisberg. 1982. *Residuals and influence in regression*. New York: Chapman and Hall.

John J Cowan and Christopher Sneden. 2006. Heavy element synthesis in the oldest stars and the early universe. *Nature*, 440(7088):1151.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring

coreferential 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 5925–5932, Hong Kong, China. Association for Computational Linguistics.

Jia Deng, R. Socher, Li Fei-Fei, Wei Dong, Kai Li, and Li-Jia Li. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 248–255.

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

Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. 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 4537–4546, Hong Kong, China. Association for Computational Linguistics.

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Ves Stoyanov, and Alexis Conneau. 2020. Self-training improves pre-training for natural language understanding.

Mengnan Du, Subhabrata Mukherjee, Yu Cheng, Milad Shokouhi, Xia Hu, and Ahmed Hassan Awadallah. 2023. Robustness challenges in model distillation and pruning for natural language understanding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1766–1778, Dubrovnik, Croatia. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917, Melbourne, Australia. Association for Computational Linguistics.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. 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. Association for Computational Linguistics.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018a. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018b. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Kawin Ethayarajh and Dan Jurafsky. 2020. Utility is in the eye of the user: A critique of NLP leaderboards. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4846–4853, Online. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Allyson Ettinger, Sudha Rao, Hal Daumé III, and Emily M. Bender. 2017. Towards linguistically generalizable NLP systems: A workshop and shared task. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 1–10, Copenhagen, Denmark. Association for Computational Linguistics.

Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. 2020. Accelerating real-time question answering via question generation. *CoRR*, abs/2009.05167.

Chrisantha Fernando, Dylan Sunil Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2024. Promptbreeder: Self-referential self-improvement via prompt evolution.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79.

Arduin Findeis, Timo Kaufmann, Eyke Hüllermeier, Samuel Albanie, and Robert Mullins. 2024. Inverse constitutional ai: Compressing preferences into principles.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of 2nd Machine Reading for Reading Comprehension (MRQA) Workshop at EMNLP*.

Deep Ganguli, Liane Lovitt, John Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Benjamin Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy

Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova Dassarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, Zachary Dodds, Tom Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, Tom B. Brown, Nicholas Joseph, Sam McCandlish, Christopher Olah, Jared Kaplan, and Jack Clark. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *ArXiv*, abs/2209.07858.

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

Matt Gardner, Jonathan Berant, Hannaneh Hajishirzi, Alon Talmor, and Sewon Min. 2019. Question answering is a format; when is it useful? *ArXiv*, abs/1909.11291.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

Atticus Geiger, Ignacio Cases, Lauri Karttunen, and Christopher Potts. 2018. Stress-

testing neural models of natural language inference with multiply-quantified sentences. *ArXiv*, abs/1810.13033.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166, Hong Kong, China. Association for Computational Linguistics.

Amir Globerson and Sam Roweis. 2006. Nightmare at test time: robust learning by feature deletion. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 353–360, New York, NY, USA. Association for Computing Machinery.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis,

Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti

Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada

Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay,

Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models.

LES Green, EC Berkeley, and C Gotlieb. 1959. Conversation with a computer. *Computers and Automation*, 8(10):9–11.

Edward Grefenstette, Robert Stanforth, Brendan O'Donoghue, Jonathan Uesato, Grzegorz Swirszcz, and Pushmeet Kohli. 2018. Strength in numbers: Trading-off robustness and computation via adversarially-trained ensembles. *CoRR*, abs/1811.09300.

Kevin Gurney. 1997. *An introduction to neural networks*. CRC press.

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

Alon Halevy, Peter Norvig, and Fernando Pereira. 2009. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3684, Florence, Italy. Association for Computational Linguistics.

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

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. 2021b. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15262–15271.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George

van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. 2022. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030. Curran Associates, Inc.

John J Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

Tom Hosking, Phil Blunsom, and Max Bartolo. 2024. Human feedback is not gold standard. In *The Twelfth International Conference on Learning Representations*.

Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020. An analysis of natural language inference benchmarks through the lens of negation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9106–9118, Online. Association for Computational Linguistics.

Hossein Hosseini, Baicen Xiao, and Radha Poovendran. 2017. Deceiving google's cloud video intelligence API built for summarizing videos. In *CVPR Workshops*, pages 1305–1309. IEEE Computer Society.

Eduard H Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. 2000. Question answering in webclopedia. In *TREC*, volume 52, pages 53–56.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense 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 2391–2401, Hong Kong, China. Association for Computational Linguistics.

William Huang, Haokun Liu, and Samuel R. Bowman. 2020. Counterfactually-augmented SNLI training data does not yield better generalization than unaugmented data. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 82–87, Online. Association for Computational Linguistics.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2023. Exploring the benefits of training expert language models over instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 14702–14729. PMLR.

Geoffrey Jefferson. 1949. The mind of mechanical man. *British Medical Journal*, 1:1105 – 1110.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. Are natural language inference models IMPPRESsive? Learning IMPlicature and

PRESupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online. Association for Computational Linguistics.

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

K Spark Jones. 2003. Is question answering a rational task. In *Questions and Answers: Theoretical and Applied Perspectives (Second CoLogNETElsNET Symposium)*, pages 24–35.

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

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st edition. Prentice Hall PTR, USA.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess,

Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Divyansh Kaushik, Douwe Kiela, Zachary C. Lipton, and Wen-tau Yih. 2021. On the efficacy of adversarial data collection for question answering: Results from a large-scale randomized study. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6618–6633, Online. Association for Computational Linguistics.

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

Daniel Khashabi, Yeganeh Kordi, and Hannaneh Hajishirzi. 2022. Unifiedqa-v2: Stronger generalization via broader cross-format training. *ArXiv*, abs/2202.12359.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Hannah Kirk, Bertie Vidgen, Paul Rottger, Tristan Thrush, and Scott Hale. 2022. Hatemoji: A test suite and adversarially-generated dataset for benchmarking and detecting emoji-based hate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1352–1368, Seattle, United States. Association for Computational Linguistics.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. 2024. The prism alignment project: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via

influence functions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR.

Jernej Kos, Ian Fischer, and Dawn Song. 2018. Adversarial examples for generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 36–42.

Venelin Kovatchev, Trina Chatterjee, Venkata S Govindarajan, Jifan Chen, Eunsol Choi, Gabriella Chronis, Anubrata Das, Katrin Erk, Matthew Lease, Junyi Jessy Li, Yating Wu, and Kyle Mahowald. 2022. longhorns at DADC 2022: How many linguists does it take to fool a question answering model? a systematic approach to adversarial attacks. In *Proceedings of the First Workshop on Dynamic Adversarial Data Collection*, pages 41–52, Seattle, WA. Association for Computational Linguistics.

A. Kumar, T. Ma, and P. Liang. 2020. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning (ICML)*.

Alexey Kurakin, Ian Goodfellow, and Sam Bengio. 2017a. Adversarial examples in the physical world. In *International Conference on Learning Representations*.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2017b. Adversarial machine learning at scale. In *International Conference on Learning Representations*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Cody Kwok, Oren Etzioni, and Daniel S Weld. 2001. Scaling question answering to the web. *ACM Transactions on Information Systems (TOIS)*, 19(3):242–262.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of the 18th International Conference on Machine Learning (ICML) 2001*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. *arXiv preprint arXiv:1704.04683*.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. 2010. *Research Methods in Human-Computer Interaction*. Wiley Publishing.

Jonathan Leader Maynard and Susan Benesch. 2016. Dangerous Speech and Dangerous Ideology: An Integrated Model for Monitoring and Prevention. *Genocide Studies and Prevention*, 9(3):70–95.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

Mina Lee, Megha Srivastava, Amelia Hardy, John Thickstun, Esin Durmus, Ashwin Paranjape, Ines Gerard-Ursin, Xiang Lisa Li, Faisal Ladhak, Frieda Rong, Rose E Wang, Minae Kwon, Joon Sung Park, Hancheng Cao, Tony Lee, Rishi Bommasani, Michael S. Bernstein, and Percy Liang. 2023. Evaluating human-language model interaction. *Transactions on Machine Learning Research*.

Douglas B Lenat and Ramanathan V Guha. 1989. *Building large knowledge-based systems; representation and inference in the Cyc project*. Addison-Wesley Longman Publishing Co., Inc.

David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *SIGIR*, pages 3–12. ACM/Springer.

Mike Lewis and Angela Fan. 2019. Generative question answering: Learning to answer the whole question. In *International Conference on Learning Representations*.

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

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

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

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

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Hugo Liu, Henry Lieberman, and Ted Selker. 2003. A model of textual affect sensing using real-world knowledge. In *Proceedings of Intelligent User Interfaces (IUI)*, pages 125–132.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019a. Inoculation by fine-tuning: A method for analyzing challenge datasets. 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 2171–2179, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Bruce P Lowerre and B Raj Reddy. 1976. Harpy, a connected speech recognition system. *The Journal of the Acoustical Society of America*, 59(S1):S97–S97.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the*

2015 Conference on Empirical Methods in Natural Language Processing, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. 2021. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking. In *Advances in Neural Information Processing Systems*, volume 34, pages 10351–10367. Curran Associates, Inc.

Kenneth Ian Manktelow. 1999. *Reasoning and thinking*. Psychology press.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. 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 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.

Mark Mazumder, Colby Banbury, Xiaozhe Yao, Bojan Karlaš, William A Gaviria Rojas, Sudnya Diamos, Greg Diamos, Lynn He, Alicia Parrish, Hannah Rose Kirk, Jessica Quaye, Charvi Rastogi, Douwe Kiela, David Jurado, David Kanter, Rafael Mosquera, Will Cukierski, Juan Ciro, Lora Aroyo, Bilge Acun, Lingjiao Chen, Mehul Smriti Raje, Max Bartolo, Sabri Eyuboglu, Amirata Ghorbani, Emmett Daniel Goodman, Addison Howard, Oana Inel, Tariq Kane, Christine Kirkpatrick, D. Sculley, Tzu-Sheng Kuo, Jonas Mueller, Tristan Thrush, Joaquin Vanschoren, Margaret Warren, Adina Williams, Serena Yeung, Newsha Ardalani, Praveen Paritosh, Ce Zhang, James Y. Zou, Carole-Jean Wu, Cody Coleman, Andrew Ng, Peter Mattson, and Vijay Janapa Reddi. 2023. Dataperf: Benchmarks

for data-centric AI development. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. The natural language decathlon: Multitask learning as question answering.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

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

Sewon Min, Victor Zhong, Richard Socher, and Caiming Xiong. 2018. Efficient and robust question answering from minimal context over documents. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1725–1735, Melbourne, Australia. Association for Computational Linguistics.

Pasquale Minervini and Sebastian Riedel. 2018. Adversarially regularising neural NLI models to integrate logical background knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 65–74, Brussels, Belgium. Association for Computational Linguistics.

Marvin L Minsky. 1969. *Semantic information processing*. The MIT Press.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. 2018. Never-ending learning. *Communications of the ACM*, 61(5):103–115.

Robert Munro, Steven Bethard, Victor Kuperman, Vicky Tzuyin Lai, Robin Melnick, Christopher Potts, Tyler Schnoebelen, and Harry Tily. 2010. Crowdsourcing and language studies: the new generation of linguistic data. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 122–130, Los Angeles. Association for Computational Linguistics.

Brad Myers, Jim Hollan, Isabel Cruz, Steve Bryson, Dick Bulterman, Tiziana Catarci, Wayne Citrin, Ephraim Glinert, Jonathan Grudin, and Yannis Ioannidis. 1996. Strategic directions in human-computer interaction. *ACM Comput. Surv.*, 28(4):794–809.

Brad A. Myers. 1998. A brief history of human-computer interaction technology. *Interactions*, 5(2):44–54.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in*

Natural Language Processing (EMNLP), pages 1953–1967, Online. Association for Computational Linguistics.

Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2010. Recognition of affect, judgment, and appreciation in text. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 806–814, Beijing, China. Coling 2010 Organizing Committee.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated MAchine Reading COmprehension dataset. *arXiv preprint arXiv:1611.09268*.

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2019. Analyzing compositionality-sensitivity of nli models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6867–6874.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Peter Norvig. 1986. Unified theory of inference for text understanding. Technical report, California University of Berkeley Graduate Division.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1):1–135.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad

discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy*, pages 372–387.

Bhargavi Paranjape, Matthew Lamm, and Ian Tenney. 2022. Retrieval-guided counterfactual generation for QA. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1670–1686, Dublin, Ireland. Association for Computational Linguistics.

Sue Taylor Parker and Kathleen Rita Gibson. 1979. A developmental model for the evolution of language and intelligence in early hominids. *Behavioral and Brain Sciences*, 2(3):367–381.

Alicia Parrish, Hannah Rose Kirk, Jessica Quaye, Charvi Rastogi, Max Bartolo, Oana Inel, Juan Ciro, Rafael Mosquera, Addison Howard, Will Cukierski, D. Sculley, Vijay Janapa Reddi, and Lora Aroyo. 2023. Adversarial nibbler: A data-centric challenge for improving the safety of text-to-image models.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037. Curran Associates, Inc.

Anselmo Peñas, Eduard H Hovy, Pamela Forner, Álvaro Rodrigo, Richard FE Sutcliffe, Corina Forascu, and Caroline Sporleder. 2011. Overview of qa4mre at clef

2011: Question answering for machine reading evaluation. In *CLEF (Notebook Papers/Labs/Workshop)*, pages 1–20.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Benjamin Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemi Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, and Jared Kaplan. 2023. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13387–13434, Toronto, Canada. Association for Computational Linguistics.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models.

In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions.

Jason Phang, Angelica Chen, William Huang, and Samuel R. Bowman. 2022. Adversarially constructed evaluation sets are more challenging, but may not be fair. In *Proceedings of the First Workshop on Dynamic Adversarial Data Collection*, pages 62–62, Seattle, WA. Association for Computational Linguistics.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2020. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In

Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2021. DynaSent: A dynamic benchmark for sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2388–2404, Online. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Grusha Prasad, Yixin Nie, Mohit Bansal, Robin Jia, Douwe Kiela, and Adina Williams. 2021. To what extent do human explanations of model behavior align with actual model behavior? In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 1–14, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jenny Preece, Yvonne Rogers, Helen Sharp, David Benyon, Simon Holland, and Tom Carey. 1994. *Human-Computer Interaction*. Addison-Wesley Longman Ltd., GBR.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified defenses against adversarial examples. In *International Conference on Learning Representations*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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

Jerome Ramos and Aldo Lipani. 2024. Extra-sharc: Explainable and scrutable reading comprehension for conversational systems. In *Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '24, page 47–56, New York, NY, USA. Association for Computing Machinery.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

A. C. Reynolds Jr. 1954. The conference on mechanical translation. *Mechanical Translation and Computational Linguistics*, 1(3):47–55.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. 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, Melbourne, Australia. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA. Association for Computational Linguistics.

Alexis Ross, Tongshuang Wu, Hao Peng, Matthew Peters, and Matt Gardner. 2022. Tailor: Generating and perturbing text with semantic controls. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3194–3213, Dublin, Ireland. Association for Computational Linguistics.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. 2021. HateCheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 41–58, Online. Association for Computational Linguistics.

Ohad Rozen, Vered Shwartz, Roee Aharoni, and Ido Dagan. 2019. Diversify your datasets: Analyzing generalization via controlled variance in adversarial datasets. In *Proceedings of the 23rd Conference on Computational Natural Language*

Learning (CoNLL), pages 196–205, Hong Kong, China. Association for Computational Linguistics.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. 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 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of natural language rules in conversational machine reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2087–2097, Brussels, Belgium. Association for Computational Linguistics.

Swarnadeep Saha, Yixin Nie, and Mohit Bansal. 2020. ConjNLI: Natural language inference over conjunctive sentences. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8240–8252, Online. Association for Computational Linguistics.

Victor Sánchez Valencia. 1991. *Studies on natural logic and categorial grammar, University of Amsterdam Ph. D*. Ph.D. thesis, thesis.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Chinnadhurai Sankar, Sandeep Subramanian, Chris Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 32–37, Florence, Italy. Association for Computational Linguistics.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Sebastian Schuster, Yuxing Chen, and Judith Degen. 2020. Harnessing the linguistic signal to predict scalar inferences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5387–5403, Online. Association for Computational Linguistics.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. 2017. The effect of different writing tasks on linguistic style: A case study of the ROC story cloze task. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 15–25, Vancouver, Canada. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *The International Conference on Learning Representations (ICLR)*.

Burr Settles. 2009. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Sasha Sheng, Amanpreet Singh, Vedanuj Goswami, Jose Magana, Tristan Thrush, Wojciech Galuba, Devi Parikh, and Douwe Kiela. 2021. Human-adversarial visual question answering. In *Advances in Neural Information Processing Systems*, volume 34, pages 20346–20359. Curran Associates, Inc.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Kurt Shuster, Da Ju, Stephen Roller, Emily Dinan, Y-Lan Boureau, and Jason Weston. 2020. The dialogue dodecathlon: Open-domain knowledge and image grounded conversational agents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2453–2470, Online. Association for Computational Linguistics.

Daniel L Silver, Qiang Yang, and Lianghao Li. 2013. Lifelong machine learning systems: Beyond learning algorithms. In *2013 AAAI spring symposium series*.

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

Robert F Simmons. 1970. Natural language question-answering systems: 1969. *Communications of the ACM*, 13(1):15–30.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergün, Ifeoma Okoh, Aisha Alaagib, Oshan Mudannayake, Zaid Alyafeai, Vu Minh Chien, Sebastian Ruder, Surya Guthikonda, Emad A. Alghamdi, Sebastian Gehrmann, Niklas Muennighoff, Max Bartolo, Julia Kreutzer, Ahmet Üstün, Marzieh Fadaee, and Sara Hooker. 2024. Aya dataset: An open-access collection for multilingual instruction tuning.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural*

Language Processing, pages 254–263, Honolulu, Hawaii. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

John F Sowa et al. 1992. Semantic networks. *Encyclopedia of artificial intelligence*, 2:1493–1511.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Johan Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, Cesar Ferri, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Christopher Waites, Christian Voigt, Christopher D Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, C. Daniel

Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodolà, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Xinyue Wang, Gonzalo Jaimovitch-Lopez, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Francis Anthony Shevlin, Hinrich Schuetze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B Simon, James Koppel, James Zheng, James Zou, Jan Kocon, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh Dhole, Kevin Gimpel, Kevin Omondi, Kory Wallace Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency,

Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros-Colón, Luke Metz, Lütfi Kerem Senel, Maarten Bosma, Maarten Sap, Maartje Ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramirez-Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael Andrew Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swedrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan Andrew Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter W Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan Le Bras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Russ Salakhutdinov, Ryan Andrew Chi, Seungjae Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel Stern Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham

Toshniwal, Shyam Upadhyay, Shyamolima Shammie Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven Piantadosi, Stuart Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsunori Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Venkatesh Ramasesh, vinay uday prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33.

David Stutz, Matthias Hein, and Bernt Schiele. 2019. Disentangling adversarial robustness and generalization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Moritz Sudhof, Andrés Gómez Emilsson, Andrew L. Maas, and Christopher Potts. 2014. Sentiment expression conditioned by affective transitions and social forces. In *Proceedings of 20th Conference on Knowledge Discovery and Data Mining*, pages 1136–1145, New York. ACM.

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

Saku Sugawara, Pontus Stenetorp, Kentaro Inui, and Akiko Aizawa. 2020. Assessing the benchmarking capacity of machine reading comprehension datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8918–8927.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Richard FE Sutcliffe, Anselmo Peñas, Eduard H Hovy, Pamela Forner, Álvaro Rodrigo, Corina Forascu, Yassine Benajiba, and Petya Osenova. 2013. Overview of qa4mre main task at clef 2013. In *CLEF (Working Notes)*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.

Wilson L Taylor. 1953. "cloze procedure": a new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433.

Dov Te'eni, Jane M. Carey, and Ping Zhang. 2005. *Human-Computer Interaction: Developing Effective Organizational Information Systems*. John Wiley & Sons, Inc., Hoboken, NJ, USA.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. The FEVER2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6, Hong Kong, China. Association for Computational Linguistics.

Tristan Thrush, Kushal Tirumala, Anmol Gupta, Max Bartolo, Pedro Rodriguez, Tariq Kane, William Gaviria Rojas, Peter Mattson, Adina Williams, and Douwe Kiela. 2022. Dynatask: A framework for creating dynamic AI benchmark tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 174–181, Dublin, Ireland. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne

Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. 2008. Accelerating the annotation of sparse named entities by dynamic sentence selection. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 30–37, Columbus, Ohio. Association for Computational Linguistics.

Alan M Turing. 1950. Computing machinery and intelligence. *Mind*, 59(236):433–460.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model.

Clara Vania, Phu Mon Htut, William Huang, Dhara Mungra, Richard Yuanzhe Pang, Jason Phang, Haokun Liu, Kyunghyun Cho, and Samuel R. Bowman. 2021. Comparing test sets with item response theory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1141–1158, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Bertie Vidgen and Leon Derczynski. 2020. Directions in Abusive Language Training Data: Garbage In, Garbage Out. *arXiv:2004.01670*, pages 1–26.

Bertie Vidgen, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott Hale, and Helen Margetts. 2019. Challenges and frontiers in abusive content detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 80–93, Florence, Italy. Association for Computational Linguistics.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682, Online. Association for Computational Linguistics.

Ellen M Voorhees, Donna K Harman, et al. 2005. *TREC: Experiment and evaluation in information retrieval*, volume 1. MIT press Cambridge.

Ellen M Voorhees and Dawn M Tice. 1999. The trec-8 question answering track evaluation. In *TREC*, volume 1999, page 82.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207. ACM.

Harm de Vries, Dzmitry Bahdanau, and Christopher Manning. 2020. Towards ecologically valid research on language user interfaces. *arXiv preprint arXiv:2007.14435*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019a. Universal adversarial triggers for attacking and analyzing NLP. 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 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019b. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Eric Wallace, Adina Williams, Robin Jia, and Douwe Kiela. 2022. Analyzing dynamic adversarial training data in the limit. In *Findings of the Association for*

Computational Linguistics: ACL 2022, pages 202–217, Dublin, Ireland. Association for Computational Linguistics.

David L Waltz. 1978. An english language question answering system for a large relational database. *Communications of the ACM*, 21(7):526–539.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32, pages 3266–3280. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. 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. Association for Computational Linguistics.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP-CoNLL*, volume 7, pages 22–32.

Yicheng Wang and Mohit Bansal. 2018. Robust machine comprehension models via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 575–581, New Orleans, Louisiana. Association for Computational Linguistics.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019. Investigating BERT's knowledge of language: Five analysis methods with NPIs. 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 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding Abuse: A Typology of Abusive Language Detection Subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84.

Peter C Wason. 1959. The processing of positive and negative information. *Quarterly Journal of Experimental Psychology*, 11(2):92–107.

Peter C Wason. 1961. Response to affirmative and negative binary statements. *British Journal of Psychology*, 52(2):133–142.

Peter Cathcart Wason and Philip Nicholas Johnson-Laird. 1972. *Psychology of reasoning: Structure and content*, volume 86. Harvard University Press.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making neural QA as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, Vancouver, Canada. Association for Computational Linguistics.

Joseph Weizenbaum. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Johannes Welbl, Pasquale Minervini, Max Bartolo, Pontus Stenetorp, and Sebastian Riedel. 2020. Undersensitivity in neural reading comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1152–1165, Online. Association for Computational Linguistics.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Guillaume Wenzek, Vishrav Chaudhary, Angela Fan, Sahir Gomez, Naman Goyal, Somya Jain, Douwe Kiela, Tristan Thrush, and Francisco Guzmán. 2021. Findings of the WMT 2021 shared task on large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 89–99, Online. Association for Computational Linguistics.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards AI-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Aaron Steven White, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2018. Lexicosyntactic inference in neural models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4717–4724, Brussels, Belgium. Association for Computational Linguistics.

Aaron Steven White, Elias Stengel-Eskin, Siddharth Vashishtha, Venkata Subrahmanyan Govindarajan, Dee Ann Reisinger, Tim Vieira, Keisuke Sakaguchi, Sheng Zhang, Francis Ferraro, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2020. The universal decompositional semantics dataset and decomp toolkit. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5698–5707, Marseille, France. European Language Resources Association.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2–3):165–210.

Yorick Wilks. 1975. A preferential, pattern-seeking, semantics for natural language inference. *Artificial intelligence*, 6(1):53–74.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Terry Winograd. 1971. Procedures as a representation for data in a computer program for understanding natural language. In *AI Technical Reports (1964 - 2004)*.

Rey Reza Wiyatno, Anqi Xu, Ousmane Dia, and Archy de Berker. 2019. Adversarial examples in modern machine learning: A review.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

William A Woods. 1978. Semantics and quantification in natural language question answering. *Advances in computers*, 17:1–87.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6707–6723, Online. Association for Computational Linguistics.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. 2023. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Advances in Neural Information Processing Systems*, volume 36, pages 78205–78226. Curran Associates, Inc.

Jasper Xian, Saron Samuel, Faraz Khoubsirat, Ronak Pradeep, Md Arafat Sultan, Radu Florian, Salim Roukos, Avirup Sil, Christopher Potts, and Omar Khattab. 2024. Prompts as auto-optimized training hyperparameters: Training best-in-class ir models from scratch with 10 gold labels.

Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 3905–3911. AAAI Press.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. 2020. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huan Xu and Shie Mannor. 2010. Robustness and generalization. *Machine Learning*, 86.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *EMNLP*, pages 2013–2018.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018a. 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. Association for Computational Linguistics.

Zhilin Yang, Saizheng Zhang, Jack Urbanek, Will Feng, Alexander Miller, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018b. Mastering the dungeon: Grounded language learning by mechanical turker descent. In *International Conference on Learning Representations*.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zihuiwen Ye, Fraser Greenlee-Scott, Max Bartolo, Phil Blunsom, Jon Ander Campos, and Matthias Gallé. 2024. Improving reward models with synthetic critiques.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome T. Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and evaluating general linguistic intelligence. *ArXiv*, abs/1901.11373.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907, Online. Association for Computational Linguistics.

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2805–2824.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. ReCoRD: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

Wei Emma Zhang, Quan Z. Sheng, and Ahoud Abdulrahmn F. Alhazmi. 2019. Generating textual adversarial examples for deep learning models: A survey. *CoRR*, abs/1901.06796.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

Zhiping Zheng. 2002. Answerbus question answering system. In *Proceedings of the second international conference on Human Language Technology Research*, pages 399–404. Morgan Kaufmann Publishers Inc.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and M. Zhou. 2017. Neural question generation from text: A preliminary study. *ArXiv*, abs/1704.01792.

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. The curse of performance instability in analysis datasets: Consequences, source, and suggestions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8215–8228, Online. Association for Computational Linguistics.