NLP Evaluation in the Time of Large Language Models

by

Alex Wang

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

September, 2022

Professor Samuel R. Bowman

Professor Kyunghyun Cho

Acknowledgements

First and foremost, I must thank my advisors, Samuel R. Bowman and Kyunghyun Cho. I truly could not imagine a better pair of advisors. Sam and Kyunghyun gave me the freedom, resources, and patience to explore my own ideas, but also support and direction when I needed it. They taught me how to manage 100+ experiments, how to clearly communicate my work, and everything in between.[1] They connected me to countless research opportunities and collaborations that helped establish me as a scientist in the field. Any success I had during my PhD, and any success I will have in the future, is due in large part to their mentorship.

I also am extremely grateful to have had excellent mentors beyond my advisors. When I was an aimless undergrad at Harvard with a budding interest in machine learning, Sasha Rush first showed me how to implement a neural network and read a research paper. When I was an intern at Stanford, Will Hamilton and Jure Leskovec helped bridge my initial interest in computational social science with NLP, converted my intern project into a workshop publication, and made me a competitive applicant for grad school applications. I have also learned a great deal from fantastic mentors during my internships, including Ellie Pavlick and Ian Tenney from Google; Mike Lewis from Meta; and Ronan Le Bras and Yejin Choi from the Allen Institute.

Leaving NYU is bittersweet as my time there has been incredibly enriching and formative. I am grateful to the CILVR and ML^2 labs for being my home for five years, and I am grateful I could learn from so many brilliant people there: Ilia Kulikov, Jason Lee, Phu Mon Htut, Cinjon Resnick,

[1] Among these things: alcohol. Sam gave me my first set of wine glasses. Cho bought me beers and criticized my beer choices.

Alex Warstadt, Nikita Nangia, Jason Phang, Ethan Perez, Mark Goldstein, David Brandfonbrener, Katrina Evtimova, Richard Pang, Angelica Chen, Vishakh Padmakumar, Adina Williams, Elman Mansimov, Kianté Brantley, Sean Welleck, Will Whitney, Alicia Parrish, Clara Vania, Katharina Kann, Thibault Fevry, Yada Pruksachatkun, Kelly Zhang, Amanpreet Singh, Iacer Calixto, Tal Linzen, João Sedoc, and He He. I give extra thanks to the latter two for serving on my dissertation committee, along with Mike Lewis and my advisors. I am also extremely appreciative of the admins at NYU who made my life much easier, especially Hong Tam, Santiago Pizzini, and Aimée-Catherine Zambrana, as well as the technical support staff, Shenglong Wang and Shirley Palma, and my friend Joey the Janitor.

Outside of NYU, I am appreciative of inspiring conversations and collaborations with many people, including Yoon Kim, Sam Wiseman, Sebastian Gehrmann, and Jeffrey Ling at Harvard; Felix Hill at DeepMind; Omer Levy and Julian Michael from the University of Washington; and Patrick Xia, Chandler May, Tom McCoy, and Najoung Kim from Johns Hopkins; and the many, many folks I worked with at the very productive JSALT 2018 workshop.

I also want to say thanks to the people that have supported me personally and that have made my time in grad school so much more enjoyable. I have been fortunate to have had the support of many friends: Frances Ding, Lily Zhang, Evan Yao, Richard Zou, Gordon Petty, Wentong Zhang, Wayne Zhao, Andrew Hess, Aaron Suduiko, Henry Johnson, Judson Woods, and Adam Gilfix. I especially want to thank the friends that have been there for me all throughout grad school, well before, and certainly well beyond: David Chang, Raynor Kuang, and Jimmy Lin. I am also indebted to the Hu family for essentially taking me in since coming to New York. Shenghua and Jane Hu have been my home away from home and a welcome getaway from the city. Their home-cooked meals and visits have greatly enriched the past five years of my life.

I am deeply grateful to Cherie Hu for being my sounding board, my editor, my life coach, and my cheerleader. Your love and support these past five years buoyed me in the low moments and elevated me during the high. I also appreciate reconnecting with my brothers, Andrew and

Aaron, throughout grad school, as we were not especially close before. Finally, I want to express my undying gratitude to my parents, who from my very beginning have encouraged me and have sacrificed for me so that I could be the best version of myself possible. I hope that I can pay forward all the good in my life that you have done for me.

Abstract

The field of natural language processing (NLP) has been dramatically impacted by the creation and proliferation of large language models that are pretrained on Internet-scale text data. These models have led to significant improvements on a myriad of NLP tasks. However, as the capabilities of these models drive up performance on existing task benchmarks, there is a critical need for evaluation metrics that are up-to-date with current models. In this dissertation, we develop NLP evaluation methodologies that benchmark and leverage pretrained language models. We first present two multi-task benchmarks for evaluating the generalization ability of NLP models and discuss the role of these benchmarks in the development of large language models. Next, we demonstrate that we can leverage the capabilities of pretrained language models to develop new automatic evaluation metrics that better measure the semantics of model-generated text. Specifically, we make use of the question answering abilities of pretrained models to evaluate the faithfulness of automatically generated summaries. Finally, we explore methods for crowdsourcing high-quality and challenging text generation data to address issues of data quality that have been surfaced by the ability of language models to replicate noise in benchmark datasets. Overall, we show that the rise of pretrained language models presents both challenges and opportunities in how we evaluate NLP systems, and that incorporating these very models into our evaluation methodologies offers a promising path forward.

Contents

List of Figures

List of Tables

1 Introduction

In standard supervised machine learning problems, we are provided with a dataset on which we train a model and another heldout dataset drawn from the same distribution as the training on which we evaluate the quality of the learned model. However, in natural language processing (NLP), where the task inputs and/or outputs are text, there has been increased interest in models that can be applied to arbitrary NLP tasks, not only the ones that match the distribution of the training data the model was trained on. Because these tasks all involve human language to some degree, the hope is that a model with some understanding of language can transfer that understanding across tasks. This hope has given rise to pretrained language models, colossal machine learning models that are trained on language modeling objectives with hundreds of billions of tokens and that generalize extremely well to downstream NLP tasks. The rapid development and proliferation of pretrained language models has had a profound impact across a variety of longstanding NLP tasks, such as text classification and question answering [Devlin et al. 2019; Radford et al. 2018; Lewis et al. 2020; Raffel et al. 2020, i.a.], and also quickly led to the development of surprising new capabilities, such as writing code [Austin et al. 2021] and solving math problems [Hendrycks et al. 2021].

The progress brought about by pretrained language models opens up a host of opportunities for the way we evaluate NLP systems. Pretrained language models have driven rapid progress on longstanding benchmark datasets for a variety of tasks, saturating them and limiting their utility. To continue to be able to accurately measure the capabilities of these models, we must develop new methods for building more challenging and higher quality evaluation data. Additionally, because their ability to generalize to downstream tasks is unmatched by previous approaches, we must create new evaluation paradigms to quantify this generalization ability and to measure their capacity to perform new skills. In conjunction, we can experiment with incorporating the emergent capabilities of these models into the evaluation methodologies themselves. In this dissertation, we explore these questions and opportunities.

First, we develop evaluation resources to measure how effectively pretrained models can adapt to different tasks. In Chapter 2, we introduce the GLUE benchmark, which measures how well NLP systems can generalize to diverse task and settings. The GLUE benchmark is a multi-task benchmark where all tasks share the same task format but some of the task training datasets consist of only a few hundred examples, necessitating knowledge transfer from some outside source to do well on the task. GLUE has enjoyed rapid adoption in the field, and the state-of-the-art on the benchmark has risen rapidly to the point of being saturated with respect to human crowdworker performance on the benchmark. In Chapter 3, we introduce the SuperGLUE benchmark, which refreshes the GLUE benchmark by selecting a set of more diverse set of task formats and more challenging set of tasks. Like the GLUE benchmark, the SuperGLUE benchmark has become one of the standard approaches for measuring model generalization in NLP.

Next, as the capabilities of NLP models based on large LMs continue to evolve, we can harness them to evaluate system capabilities in new ways and more robustly than before. In Chapter 4, we explore the use of pretrained language models in detecting hallucinations in the outputs of neural text summarization systems. While the outputs of text generation models have become highly fluent, they frequently contain inconsistencies with respect to previous tokens they have generated or to the input document they are conditioned on. These hallucinations or inconsistencies are a major barrier to the usability and reliability of text generation systems, but existing evaluation metrics for NLG systems are insensitive to these types of errors. We decompose the problem of detecting these hallucinations into one of question answering and question generation, two problems on which pretrained language models have made significant progress in recent years. We demonstrate that evaluation metrics based on pretrained language models are substantially better at detecting hallucinations in generated text compared to existing evaluation metrics. The success of the method is indicative of the promise of leveraging pretrained models to measure a variety of properties of generated text that was beyond the scope of the previous generation of evaluation metrics.

Finally, as system capabilities evolve, our evaluation data need to similarly evolve, as older datasets saturate and more powerful models are better able to exploit unexpected biases and expose issues in the evaluation data. Sourcing and maintaining high data quality for text generation tasks is particularly challenging as text generation datasets typically rely on finding naturally occurring data sources or using heuristics to massage data sources into the task format. In the context of summarization, this approach to dataset creation has led to benchmark datasets that contain a myriad of problems that are picked up by models trained on the datasets [Kryscinski et al. 2019]. In Chapter 5, we explore alternate methods for creating high-quality test sets for natural language generation tasks. In particular, we focus on crowdsourcing summaries for short stories in a cost-efficient manner. The resulting dataset SQuALITY is a high-quality, multi-reference summarization dataset that is beyond the capabilities of existing summarization models.

Altogether, we present a overview of the evaluation landscape of NLP models in the era of large language models, including methods for evaluating such models and for incorporating these models into evaluation methodologies. We conclude in Chapter 6 with a discussion of open problems and opportunities for further research in robust evaluation of modern NLP systems.

1.1 List of Contributions

- **Wang, Alex**, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding." EMNLP 2018.

Citation: Wang et al. [2019b]

I aggregated and prepared the data for the benchmark tasks. I also developed baselines for the benchmark tasks and ran all the experiments. Finally, I helped Amanpreet test the benchmark website evaluation and prepared the data for release. Additionally, I worked with the other authors in writing the paper.

- **Wang, Alex**, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. "SuperGLUE: A stickier benchmark for general-purpose language understanding systems." NeurIPS 2019.

  Citation: Wang et al. [2019a]

  I helped develop guidelines for new tasks to SuperGLUE with the other authors. Like with GLUE, I led the experiments for testing candidate tasks, helped test the benchmark website evaluation, prepared the data for release, and assisted in the writing the paper.

- **Wang, Alex**, Kyunghyun Cho, and Mike Lewis. "Asking and Answering Questions to Evaluate the Factual Consistency of Summaries." ACL 2020.

  Citation: Wang et al. [2020]

  I developed the high-level idea with Kyunghyun and Mike. I led the main experiments and analysis, as well as the human evaluation. Mike and Kyunghyun provided guidance and helped with the writing.

- **Wang, Alex**, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R. Bowman. "SQuALITY: A Quality Dataset for Question-Focused Summarization." In preparation.

  Citation: Wang et al. [2022]

  I developed the high-level idea and crowdsourcing protocol with feedback from Sam. I conducted the crowdsourcing protocol by adapting existing infrastructure Angie and data curated by Richard. I developed baselines on the dataset, aided by Jason. Sam provided high-level guidance throughout the project.

2 | GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding

2.1 Introduction

The human ability to understand language is *general*, *flexible*, and *robust*. In contrast, most NLU models above the word level are designed for a specific task and struggle with out-of-domain data. If we aspire to develop models with understanding beyond the detection of superficial correspondences between inputs and outputs, then it is critical to develop a more unified model that can learn to execute a range of different linguistic tasks in different domains.

To facilitate research in this direction, we present the General Language Understanding Evaluation (GLUE, `gluebenchmark.com`) benchmark: a collection of NLU tasks including question answering, sentiment analysis, and textual entailment, and an associated online platform for model evaluation, comparison, and analysis. GLUE does not place any constraints on model architecture beyond the ability to process single-sentence and sentence-pair inputs and to make corresponding predictions. For some GLUE tasks, training data is plentiful, but for others it is limited or fails to match the genre of the test set. GLUE therefore favors models that can learn to represent linguistic knowledge in a way that facilitates sample-efficient learning and effective knowledge-transfer across tasks. While none of the datasets in GLUE were created from scratch for the benchmark, four of them feature privately-held test data, which will be used to ensure that the benchmark is used fairly.

To understand the types of knowledge learned by models and to encourage linguistic or semantically-meaningful solution strategies, GLUE also includes a set of hand-crafted analysis examples for probing trained models. This dataset is designed to highlight common phenomena, such as the use of world knowledge, logical operators, and lexical entailments, that models must grasp if they are to robustly solve the tasks.

To better understand the challenged posed by GLUE, we conduct experiments with simple baselines and state-of-the-art sentence representation models. We find that unified multi-task trained models slightly outperform comparable models trained on each task separately. Our best multi-task model makes use of ELMo [Peters et al. 2018], a recently proposed pre-training technique. However, this model still achieves a fairly low absolute score, indicating room for improved general NLU systems. Analysis with our diagnostic dataset reveals that our baseline models deal well with strong lexical signals but struggle with deeper logical structure.

In summary, we offer: (i) A suite of nine sentence or sentence-pair NLU tasks, built on established annotated datasets and selected to cover a diverse range of text genres, dataset sizes, and degrees of difficulty. (ii) An online evaluation platform and leaderboard, based primarily on privately-held test data. The platform is model-agnostic, and can evaluate any method capable of producing results on all nine tasks. (iii) An expert-constructed diagnostic evaluation dataset. (iv) Baseline results for several major existing approaches to sentence representation learning.

2.2 Related Work

Collobert et al. [2011], one of the earliest works exploring deep learning for NLP, used a multi-task model with a shared sentence understanding component to jointly learn POS tagging, chunking,

Corpus	|Train|	|Dev|	|Test|	Task	Metrics	Domain
Single-Sentence Tasks						
CoLA	8.5k	1k	**1k**	acceptability	Matthews corr.	misc.
SST-2	67k	872	1.8k	sentiment	acc.	movie reviews
Similarity and Paraphrase Tasks						
MRPC	3.7k	408	1.7k	paraphrase	acc./F1	news
STS-B	7k	1.5k	1.4k	sentence similarity	Pearson/Spearman corr.	misc.
QQP	364k	40k	**391k**	paraphrase	acc./F1	Quora
Inference Tasks						
MNLI	393k	20k	**20k**	NLI	matched/mismatched acc.	misc.
QNLI	108k	5.7k	5.7k	QA/NLI	acc.	Wikipedia
RTE	2.5k	276	3k	NLI	acc.	misc.
WNLI	634	71	**146**	coreference/NLI	acc.	books

Table 2.1: Task descriptions and statistics. All tasks are single sentence or sentence pair classification, except STS-B, which is a regression task. MNLI has three classes; all other classification tasks have two. Test sets shown in bold use labels that have never been made public in any form.

named entity recognition, and semantic role labeling. More recent work has explored using labels from core NLP tasks to supervise training of lower levels of deep neural networks [Søgaard and Goldberg 2016; Hashimoto et al. 2016] and automatically learning cross-task sharing mechanisms for multi-task learning [Ruder et al. 2017].

Beyond multi-task learning, much work towards developing general NLU systems has focused on sentence-to-vector encoder functions [Le and Mikolov 2014; Kiros et al. 2015, i.a.], leveraging unlabeled data [Hill et al. 2016; Peters et al. 2018], labeled data [Conneau and Kiela 2018; McCann et al. 2017], and combinations of these [Collobert et al. 2011; Subramanian et al. 2018]. In this line of work, a standard evaluation practice has emerged, recently codified as SentEval [Conneau et al. 2017; Conneau and Kiela 2018]. Like GLUE, SentEval relies on a set of existing classification tasks that involve either one or two sentences as inputs. Unlike GLUE, SentEval only evaluates sentence-to-vector encoders. Specifically, SentEval feeds the output of a pre-trained sentence encoder into lightweight task-specific models (typically linear classifiers) that are trained and tested on task-specific data.

SentEval is well-suited for evaluating sentence representations *in isolation*. However, cross-sentence contextualization and alignment, such as that yielded by methods like soft-attention, is instrumental in achieving state-of-the-art performance on tasks such as machine translation [Bahdanau et al. 2014; Vaswani et al. 2017], question answering [Seo et al. 2016; Xiong et al. 2016], and natural language inference [Rocktäschel et al. 2016] . GLUE is designed to facilitate the development of these methods: it is model-agnostic, allowing for any kind of representation or contextualization, including models that use no systematic vector or symbolic representations for sentences whatsoever. Indeed, among the baseline models we evaluate, the use of attention consistently leads to improved performance on GLUE.

GLUE also diverges from SentEval in the selection of evaluation tasks that are included in the suite. Many of the SentEval tasks are closely related to sentiment analysis, such as MR [Pang and Lee 2005], SST [Socher et al. 2013], CR [Hu and Liu 2004], and SUBJ [Pang and Lee 2004]. Other tasks are so close to being solved that evaluation on them is relatively uninformative, such as MPQA [Wiebe et al. 2005] and TREC question classification [Voorhees et al. 1999]. In GLUE, we attempt to construct a benchmark that is both diverse and difficult.

In work which appeared after the initial launch of GLUE, McCann et al. [2018] introduce decaNLP, which also scores NLP systems based on their performance on multiple datasets. Their benchmark recasts the ten evaluation tasks as question answering, converting tasks like summarization and text-to-SQL semantic parsing into question answering using automatic transformations. That benchmark lacks the leaderboard and error analysis toolkit of GLUE, but more importantly, we see it as pursuing a more ambitious but less immediately practical goal: While GLUE rewards methods that yield good performance on a circumscribed set of tasks using methods like those that are currently used for those tasks, their benchmark rewards systems that make progress toward their goal of unifying all of NLU under the rubric of question answering.

2.3 Tasks

GLUE is centered on nine English sentence understanding tasks, which cover a broad range of domains, data quantities, and difficulties. As the goal of GLUE is to spur development of generalizable NLU systems, we design the benchmark such that good performance should require a model to share substantial knowledge (e.g., trained parameters) across all tasks, while still maintaining some task-specific components. Though it is possible to train a single model for each task and evaluate the resulting set of models on this benchmark, we expect that our inclusion of several data-scarce tasks will ultimately render this approach uncompetitive. We describe the tasks below and in Table 2.1. Appendix A.1.1 includes additional details. Unless otherwise mentioned, tasks are evaluated on accuracy and are balanced across classes.

2.3.1 Single-Sentence Tasks

CoLA The Corpus of Linguistic Acceptability [Warstadt et al. 2019] consists of English acceptability judgments drawn from books and journal articles on linguistic theory. Each example is a sequence of words annotated with whether it is a grammatical English sentence. Judgments of this particular kind are the primary form of evidence in syntactic theory [Schütze 1996], so a machine learning system capable of predicting them reliably would offer potentially substantial evidence on questions of language learnability and innate bias. Following the authors, we use the Matthews correlation coefficient [Matthews 1975] as the evaluation metric, which evaluates classifiers on unbalanced binary classification and ranges from -1 to 1, with 0 being the performance of uninformed guessing. We use the standard test set, for which we obtained private labels from the authors. We report a single performance number on the combination of the in- and out-of-domain sections of the test set.

SST-2 The Stanford Sentiment Treebank [Socher et al. 2013] consists of sentences extracted from movie reviews and human annotations of their sentiment. Given a sentence, the task is to determine the sentiment of the sentence. We use the two-way (positive/negative) class split, and use only sentence-level labels.

2.3.2 Similarity and Paraphrase Tasks

MRPC The Microsoft Research Paraphrase Corpus [Dolan and Brockett 2005] is a corpus of sentence pairs automatically extracted from online news sources, with human annotations for whether the sentences in the pair are semantically equivalent. Because the classes are imbalanced (68% positive, 32% negative), we follow common practice and report both accuracy and F1 score.

QQP The Quora Question Pairs[1] dataset is a collection of question pairs from the community question-answering website Quora. Given two questions, the task is to determine whether they are semantically equivalent. As in MRPC, the class distribution in QQP is unbalanced (37% positive, 63% negative), so we report both accuracy and F1 score. We use the standard test set, for which we obtained private labels from the authors.

STS-B The Semantic Textual Similarity Benchmark [Cer et al. 2017] is a collection of sentence pairs drawn from news headlines, video and image captions, and natural language inference data. Each pair is human-annotated with a similarity score from 1 to 5; the task is to predict these scores. Follow common practice, we evaluate using Pearson and Spearman correlation coefficients.

2.3.3 Inference Tasks

MNLI The Multi-Genre Natural Language Inference Corpus [Williams et al. 2018] is a crowd-sourced collection of sentence pairs with textual entailment annotations. Given a premise sen-

[1] data.quora.com/First-Quora-Dataset-Release-Question-Pairs

tence and a hypothesis sentence, the task is to predict whether the premise entails the hypothesis (*entailment*), contradicts the hypothesis (*contradiction*), or neither (*neutral*). The premise sentences are gathered from ten different domains of text, including transcribed speech, fiction, and government reports. We use the standard test set, for which we obtained private labels from the authors, and evaluate on both the *matched* (in-domain) and *mismatched* (cross-domain) sections. We also use and recommend the SNLI corpus [Bowman et al. 2015] as 550k examples of auxiliary training data.

QNLI The Stanford Question Answering Dataset (Rajpurkar et al. 2016) is a question-answering dataset consisting of question-paragraph pairs, where one of the sentences in the paragraph (drawn from Wikipedia) contains the answer to the corresponding question (written by an annotator). We convert the task into sentence pair classification by forming a pair between each question and each sentence in the corresponding context, and filtering out pairs with low lexical overlap between the question and the context sentence. The task is to determine whether the context sentence contains the answer to the question. This modified version of the original task removes the requirement that the model select the exact answer, but also removes the simplifying assumptions that the answer is always present in the input and that lexical overlap is a reliable cue. This process of recasting existing datasets into NLI is similar to methods introduced in White et al. [2017]. We call the converted dataset QNLI (Question-answering NLI).

RTE The Recognizing Textual Entailment (RTE) datasets come from a series of annual challenges on the task of textual entailment. We combine the data from RTE1 [Dagan et al. 2006], RTE2 [Bar Haim et al. 2006], RTE3 [Giampiccolo et al. 2007], and RTE5 [Bentivogli et al. 2009].[2] Examples are constructed based on news and Wikipedia text. We convert all datasets to a two-class split, where for three-class datasets we collapse *neutral* and *contradiction* into *not_entailment*, for consistency.

[2] RTE4 is not publicly available, while RTE6 and RTE7 do not fit the standard NLI task.

Tags	Sentence 1	Sentence 2	Fwd	Bwd
Lexical Entailment (Lexical Semantics), Downward Monotone (Logic)	The timing of the meeting has not been set, according to a Starbucks spokesperson.	The timing of the meeting has not been considered, according to a Starbucks spokesperson.	N	E
Universal Quantifiers (Logic)	Our deepest sympathies are with all those affected by this accident.	Our deepest sympathies are with a victim who was affected by this accident.	E	N
Quantifiers (Lexical Semantics), Double Negation (Logic)	I have never seen a hummingbird not flying.	I have never seen a hummingbird.	N	E

Table 2.2: Examples from the diagnostic set. *Fwd* denotes the label when sentence 1 is the premise; *Bwd* is the label when sentence 2 is the premise. Labels are *entailment* (E), *neutral* (N), or *contradiction* (C). Examples are tagged with the phenomena they demonstrate, and each phenomenon belongs to one of four broad categories (in parentheses). See Table A.1 in Appendix A.1.1 for a complete tag taxonomy.

WNLI The Winograd Schema Challenge [Levesque et al. 2012] is a reading comprehension task in which a system must read a sentence with a pronoun and select the referent of that pronoun from a list of choices. The examples are manually constructed to foil simple statistical methods: Each one is contingent on contextual information provided by a single word or phrase in the sentence. To convert the problem into sentence pair classification, we construct sentence pairs by replacing the ambiguous pronoun with each possible referent. The task is to predict if the sentence with the pronoun substituted is entailed by the original sentence. We use a small evaluation set consisting of new examples derived from fiction books[3] that was shared privately by the authors of the original corpus. While the included training set is balanced between two classes, the test set is imbalanced between them (35% entailment, 65% not entailment). As with QNLI, each example is evaluated separately, so there is not a systematic correspondence between a model's score on this task and its score on the unconverted original task. We call converted dataset WNLI (Winograd NLI).

[3] See similar examples at `cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WS.html`

2.3.4 Evaluation

The GLUE benchmark follows the same evaluation model as SemEval and Kaggle. To evaluate a system on the benchmark, one must run the system on the provided test data for the tasks, then upload the results to the website for scoring. The benchmark site then shows per-task scores, as well as a macro-average of those scores to determine a system's position on the leaderboard. For tasks with multiple metrics (e.g., accuracy and F1), we use an unweighted average of the metrics as the score for the task when computing the overall macro-average. The website also provides fine- and coarse-grained results on the diagnostic dataset. See Appendix A.1.3 for details.

2.3.5 Data and Bias

We do not endorse the use of the task training sets for any specific *non-research* use. They do not cover every dialect of English one may wish to handle, nor languages other than English. As all of them contain text or annotations that were collected in uncontrolled settings, they contain evidence of stereotypes and biases that one may not wish one's system to learn [Rudinger et al. 2017].

2.4 Diagnostic Dataset

Drawing inspiration from the FraCaS suite [Cooper et al. 1996] and the recent Build-It-Break-It competition [Ettinger et al. 2017], we include a small, manually-curated test set (with private labels) for the analysis of system performance. While the main benchmark mostly reflects an application-driven distribution of examples, our diagnostic dataset highlights a pre-defined set of modeling-relevant phenomena.

Each example in the diagnostic dataset is an NLI sentence pair with fine-grained tags for the phenomena it demonstrates. The NLI task is well-suited to this kind of analysis, as it can straightforwardly evaluate the full set of skills involved in (ungrounded) sentence understanding, from the resolution of syntactic ambiguity to pragmatic reasoning with world knowledge. We ensure that the data is reasonably diverse by producing examples for a wide variety of linguistic phenomena, and basing our examples on naturally-occurring sentences from several domains. This approaches differs from that of FraCaS, which was designed to test linguistic theories with a minimal and uniform set of examples. A sample from our dataset is shown in Table 2.2, and a full list of linguistic categories is in Table A.1 in the appendix.

Domains We construct sentence pairs based on text from four domains: News (articles linked from the front page), Reddit (threads linked from the Front Page), Wikipedia (Featured Articles), and academic papers from recent ACL conferences. We include 100 sentence pairs constructed from each source and 150 artificially-constructed sentence pairs for 550 total.

Annotation Process We begin with a target set of phenomena, based roughly on those used in the FraCaS suite [Cooper et al. 1996]. We construct each example by locating a sentence that can be easily made to demonstrate a target phenomenon, and editing it in two ways to produce an appropriate sentence pair. We make minimal modifications so as to maintain high lexical and structural overlap within each sentence pair and limit superficial cues. We then label the inference relationships between the sentences, considering each sentence alternatively as the premise, producing two labeled examples for each pair (1100 total). Where possible, we produce several pairs with different labels for a single source sentence, to have minimal sets of sentence pairs that are lexically and structurally very similar but correspond to different entailment relationships. The resulting labels are 42% *entailment*, 35% *neutral*, and 23% *contradiction*.

Evaluation Since the class distribution in the diagnostic set is not balanced, we use R_3 [Gorodkin 2004], a three-class generalization of the Matthews correlation coefficient, for evaluation.

In light of recent work showing that crowdsourced data often contains artifacts which can be exploited to perform well without solving the intended task [Schwartz et al. 2017; Gururangan et al. 2018; Poliak et al. 2018b; Tsuchiya 2018], we audit the data for such artifacts. We reproduce the methodology of Gururangan et al. [2018], training two fastText classifiers [Joulin et al. 2017] to predict entailment labels on SNLI and MNLI using only the hypothesis as input. Testing the trained classifiers on the diagnostic data, we obtain accuracies close to chance, 32.7% and 36.4% respectively, showing that the data does not suffer from artifacts of this kind.

To establish human baseline performance on the diagnostic set, we have six NLP researchers annotate 50 sentence pairs (100 entailment examples) randomly sampled from the diagnostic set. Inter-annotator agreement is high, with a Fleiss's κ of 0.73. The average R_3 score among the annotators is 0.80, much higher than any of the baseline systems described in Section 2.5.

Intended Use Because these analysis examples are hand-picked to address certain phenomena, we expect that they will not be representative of the distribution of language as a whole, even in the targeted domains. However, NLI is a task with no natural input distribution. We deliberately select sentences that we hope will be able to provide insight into what models are doing, what phenomena they catch on to, and where are they limited. This means that the raw performance numbers on the analysis set should be taken with a grain of salt. The set is provided not as a benchmark, but as an analysis tool to paint in broad strokes the kinds of phenomena a model may or may not capture, and to provide a set of examples that can serve for error analysis, qualitative model comparison, and development of adversarial examples that expose a model's weaknesses.

2.5 Baselines

We evaluate a simple multi-task learning model trained on the benchmark tasks, as well as several more sophisticated variants based on recent pre-training methods, as baselines. We briefly describe them here. See Appendix A.1.2 for details. We implement our models in the AllenNLP

		Single Sentence		**Similarity and Paraphrase**			**Natural Language Inference**			
Model	**Avg**	**CoLA**	**SST-2**	**MRPC**	**QQP**	**STS-B**	**MNLI**	**QNLI**	**RTE**	**WNLI**
Single-Task Training										
BiLSTM	62.0	15.7	85.9	69.3/79.4	81.7/61.4	66.0/62.8	70.3/70.8	60.8	52.8	62.3
+ELMo	66.2	**35.0**	90.2	69.0/80.8	85.7/65.6	64.0/60.2	72.9/73.4	69.4	50.1	**65.1**
+CoVe	62.4	14.5	88.5	73.4/81.4	83.3/59.4	67.2/64.1	64.5/64.8	64.8	53.5	61.6
+Attn	60.0	15.7	85.9	68.5/80.3	83.5/62.9	59.3/55.8	74.2/73.8	51.9	51.9	55.5
+Attn, ELMo	64.8	**35.0**	90.2	68.8/80.2	**86.5**/**66.1**	55.5/52.5	**76.9**/**76.7**	61.1	50.4	**65.1**
+Attn, CoVe	60.8	14.5	88.5	68.6/79.7	84.1/60.1	57.2/53.6	71.6/71.5	53.8	52.7	64.4
Multi-Task Training										
BiLSTM	63.5	24.0	85.8	71.9/82.1	80.2/59.1	68.8/67.0	65.8/66.0	71.1	46.8	63.7
+ELMo	64.8	27.5	89.6	76.2/83.5	78.5/57.8	67.0/65.9	67.1/68.0	66.7	55.7	62.3
+CoVe	62.2	16.2	84.3	71.8/80.0	82.0/59.1	68.0/67.1	65.3/65.9	70.4	44.2	**65.1**
+Attn	65.7	0.0	85.0	75.1/**83.7**	84.3/63.6	73.9/71.8	72.2/72.1	82.1	**61.7**	63.7
+Attn, ELMo	**69.0**	18.9	**91.6**	**77.3**/83.5	85.3/63.3	72.8/71.1	75.6/75.9	81.7	61.2	**65.1**
+Attn, CoVe	64.3	19.4	83.6	75.2/83.0	84.9/61.1	72.3/71.1	69.9/68.7	78.9	38.3	**65.1**
Pre-Trained Sentence Representation Models										
CBoW	58.9	0.0	80.0	73.4/81.5	79.1/51.4	61.2/58.7	56.0/56.4	75.1	54.1	62.3
Skip-Thought	61.5	0.0	81.8	71.7/80.8	82.2/56.4	71.8/69.7	62.9/62.8	74.7	53.1	**65.1**
InferSent	64.7	4.5	85.1	74.1/81.2	81.7/59.1	75.9/75.3	66.1/65.7	79.8	58.0	**65.1**
DisSent	62.1	4.9	83.7	74.1/81.7	82.6/59.5	66.1/64.8	58.7/59.1	75.2	56.4	**65.1**
GenSen	66.6	7.7	83.1	76.6/83.0	82.9/59.8	**79.3**/**79.2**	71.4/71.3	**82.3**	59.2	**65.1**

Table 2.3: Baseline performance on the GLUE tasks. For MNLI, we report accuracy on the matched and mismatched test sets. For MRPC and Quora, we report accuracy and F1. For STS-B, we report Pearson and Spearman correlation. For CoLA, we report Matthews correlation. For all other tasks we report accuracy. All values are scaled by 100. A similar table is presented on the online platform.

library [Gardner et al. 2017].

Architecture Our simplest baseline architecture is based on sentence-to-vector encoders, and sets aside GLUE's ability to evaluate models with more complex structures. Taking inspiration from Conneau et al. [2017], the model uses a two-layer, 1500D (per direction) BiLSTM with max pooling and 300D GloVe word embeddings [840B Common Crawl version; Pennington et al. 2014]. For single-sentence tasks, we encode the sentence and pass the resulting vector to a classifier. For sentence-pair tasks, we encode sentences independently to produce vectors u, v, and pass $[u; v; |u - v|; u * v]$ to a classifier. The classifier is an MLP with a 512D hidden layer.

We also consider a variant of our model which for sentence pair tasks uses an attention mechanism inspired by Seo et al. [2016] between all pairs of words, followed by a second BiLSTM with max pooling. By explicitly modeling the interaction between sentences, these models fall outside the sentence-to-vector paradigm.

Pre-Training We augment our base model with two recent methods for pre-training: ELMo and CoVe. We use existing trained models for both.

ELMo uses a pair of two-layer neural language models (one forward, one backward) trained on the Billion Word Benchmark [Chelba et al. 2013]. Each word is represented by a contextual embedding, produced by taking a linear combination of the corresponding hidden states of each layer of the two models. We follow the authors' recommendations[4] and use ELMo embeddings in place of any other embeddings.

CoVe [McCann et al. 2017] uses a sequence-to-sequence model with a two-layer BiLSTM encoder trained for English-to-German translation. The CoVe vector of a word is the corresponding hidden state of the top-layer LSTM. As in the original work, we concatenate the CoVe vectors to the GloVe word embeddings.

Training We train our models with the BiLSTM sentence encoder and post-attention BiLSTMs shared across tasks, and classifiers trained separately for each task. For each training update, we sample a task to train with a probability proportional to the number of training examples for each task. We train our models with Adam [Kingma and Ba 2014] with initial learning rate 10^{-3} and batch size 128. We use the macro-average score as the validation metric and stop training when the learning rate drops below 10^{-5} or performance does not improve after 5 validation checks.

We also train a set of single-task models, which are configured and trained identically, but share no parameters. While this is generally an effective model for the tasks under study, to allow for fair comparisons with the multi-task analogs we do not tune parameter or training settings for each task, so these single-task models do not generally represent the state of the art for each task.

[4] github.com/allenai/allennlp/blob/master/tutorials/how_to/elmo.md

Sentence Representation Models Finally, we evaluate the following trained sentence-to-vector encoder models using our benchmark: average bag-of-words using GloVe embeddings (CBoW), Skip-Thought [Kiros et al. 2015], InferSent [Conneau et al. 2017], DisSent [Nie et al. 2017], and GenSen [Subramanian et al. 2018]. See Appendix A.1.2 for additional details. For these models, we only train task-specific classifiers on the representations they produce.

2.6 Benchmark Results

We train three runs of each model and evaluate the run with the best macro-average development set performance. For single-task and sentence representation models, we evaluate the best run for each individual task. We present performance on the main benchmark tasks in Table 2.3.

In most cases, using multi-task training over single-task training yields better overall scores, particularly among the parameter-rich attention models. Attention generally hurts performance in single task training, but helps in multi-task training. We see a consistent improvement in using ELMo embeddings in place of GloVe or CoVe embeddings, particularly for single-sentence tasks. Using CoVe slightly improves on GloVe for single task training but not for multi-task training.

Among the pre-trained sentence representation models, we observe fairly consistent gains by moving from CBoW to Skip-Thought to Infersent and GenSen. Relative to the models trained directly on the GLUE tasks, InferSent is competitive and GenSen outperforms all but the two best.

Looking at results per task, we find that the sentence representation models substantially underperform on CoLA compared to the models directly trained on the task. Similarly, with the exception of InferSent, the sentence representation models are outperformed on SST by our BiLSTM and its non-CoVe variants. These discrepancies indicate a need for better transfer methods for generalizing outside of the tasks a model was trained on and for task diversity in evaluation methods, as we have sought to do with GLUE. On the other hand, for STS-B, there is a significant gap between the models trained directly on the task and the best sentence representation model,

		Coarse-Grained				**Fine-Grained**					
Model	**All**	**LS**	**PAS**	**L**	**K**	**UQuant**	**MNeg**	**2Neg**	**Coref**	**Restr**	**Down**
Single-Task Training											
BiLSTM	21	25	24	16	16	70	53	4	21	-15	**12**
+ELMo	20	20	21	14	17	70	20	**42**	33	-26	-3
+CoVe	21	19	23	20	18	71	47	-1	33	-15	8
+Attn	25	24	30	20	14	50	47	21	**38**	-8	-3
+Attn, ELMo	**28**	**30**	**35**	**23**	14	**85**	20	**42**	33	-26	-3
+Attn, CoVe	24	29	29	18	12	77	50	1	18	-1	**12**
Multi-Task Training											
BiLSTM	19	16	22	16	17	71	35	-8	26	**0**	8
+ELMo	19	15	21	17	**21**	70	**60**	15	26	**0**	**12**
+CoVe	17	15	21	14	16	50	31	-8	25	-15	**12**
+Attn	25	23	32	19	16	58	26	-5	28	-1	-20
+Attn, ELMo	23	24	30	17	13	78	27	37	30	-15	-20
+Attn, CoVe	20	16	25	15	17	78	37	14	31	-15	8
Pre-Trained Sentence Representation Models											
CBoW	9	6	13	5	10	3	0	13	28	-15	-11
Skip-Thought	12	2	23	11	9	61	6	-2	30	-15	0
InferSent	18	20	20	15	14	77	50	-20	15	-15	-9
DisSent	16	16	19	13	15	70	43	-11	20	-36	-09
GenSen	20	28	26	14	12	78	57	2	21	-15	**12**

Table 2.4: Results on the diagnostic set. We report R_3 coefficients between gold and predicted labels, scaled by 100. The coarse-grained categories (left) are *Lexical Semantics* (**LS**), *Predicate-Argument Structure* (**PAS**), *Logic* (**L**), and *Knowledge and Common Sense* (**K**). Our example fine-grained categories (right) are *Universal Quantification* (**UQuant**), *Morphological Negation* (**MNeg**), *Double Negation* (**2Neg**), *Anaphora/Coreference* (**Coref**), *Restrictivity* (**Restr**), and *Downward Monotone* (**Down**).

which we interpret as indicating the necessity of using transfer learning methods trained on data outside of the GLUE benchmark in order to solve it. Finally, there are tasks for which no model does particularly well. On WNLI, no model exceeds most-frequent-class guessing (65.1%). On RTE and in aggregate, even our best baselines leave room for improvement. These early results indicate that solving GLUE is beyond the capabilities of current models and methods, and that training on auxiliary tasks seems a necessary and promising direction.

2.7 Analysis

We analyze the baselines by evaluating each model's MNLI classifier on the diagnostic set to get a better sense of their linguistic capabilities. Results are presented in Table 2.4.

Coarse Categories Overall performance is low for all models: The highest total score of 28 still denotes poor absolute performance. Performance tends to be higher on Predicate-Argument Structure and lower on Knowledge, though numbers are not closely comparable across categories. Unlike on the main benchmark, the multi-task models are almost always outperformed by their single-task counterparts. This is perhaps unsurprising, since with our simple multi-task training regime, there is likely some destructive interference between MNLI and the other tasks. The models trained on the GLUE tasks largely outperform the pretrained sentence representation models, with the exception of GenSen. Using attention has a greater influence on diagnostic scores than using ELMo or CoVe, which we take to indicate that attention is especially important for generalization in NLI.

Fine-Grained Subcategories Most models handle universal quantification relatively well. Looking at relevant examples, it seems that catching on to lexical cues such as "all" often suffices for good performance. Similarly, lexical cues often provide good signal in examples of morphological negation.

We also observe weaknesses that vary between models. Double negation is especially difficult for the GLUE-trained models that only use GloVe embeddings. This is ameliorated by ELMo, and to some degree CoVe, perhaps because the translation and language modeling objectives teach models that phrases like "not bad" and "okay" have similar distributions. Also, while attention improves overall results, attention models tend to struggle with downward monotonicity. Examining their predictions, we found that the models were sensitive to hypernym/hyponym substitutions as signals of entailment, but predicted it in the wrong direction (as if the substituted word was in an upward monotone context). Restrictivity examples, which often depend on nuances of quantifier scope, are especially difficult for all models.

Overall, there is evidence that going beyond sentence-to-vector representations, e.g. with an attention mechanism, might aid performance on out-of-domain data, and that transfer methods like ELMo and CoVe encode linguistic information specific to their supervision signal. However, increased representational capacity may lead to overfitting, such as the failure of attention models in downward monotone contexts. We expect that our platform and diagnostic dataset will be useful for similar analyses in the future, so that model designers can better understand their models' generalization behavior and implicit knowledge.

2.8 Conclusion

We introduce GLUE, a platform and collection of resources for evaluating and analyzing natural language understanding systems. We find that, in aggregate, models trained jointly on our tasks see better performance than the combined performance of models trained for each task separately. We confirm the utility of attention mechanisms and transfer learning methods such as ELMo in NLU systems, which combine to outperform the best sentence representation models on the GLUE benchmark, but still leave room for improvement. When evaluating these models on our diagnostic dataset, we find that they fail (often spectacularly) on many linguistic phenomena, suggesting possible avenues for future work. In sum, the question of how to design general-purpose NLU models remains unanswered, and we believe that GLUE can provide fertile soil for addressing this challenge.

2.9 Retrospective

In Chapter 2, we introduced the GLUE benchmark, a standardized evaluation protocol for multi-task NLP based on a set of challenging and diverse NLP tasks. The benchmark incentivizes knowledge transfer between tasks or from pretraining by including tasks with small training datasets and a uniform interface across tasks (sentence or sentence-pair classification (and regression)). This common task interface also improves the usability of the benchmark by allowing users to avoid needing to create many task-specific models.

At the time of release, our best performing baseline used the pretrained model ELMo [Peters et al. 2018]. ELMo uses two unidirectional language models (one left-to-right and one right-to-left) to compute contextual word embeddings. The language models are pretrained on the Billion Word Benchmark [Chelba et al. 2013], a large unlabeled text corpus. Shortly after the release of GLUE, GPT [Radford et al. 2018] substantially improved upon ELMo predominantly by scaling up the language model pretraining with larger models, more training data, and longer training time. Since then, scaling up language modeling pretraining has been the most consistently effective way to progress on the GLUE benchmark and to learn more generalizable NLU models. Pretrained language models such as BERT [Devlin et al. 2019], GPT2 [Radford et al. 2019], RoBERTa [Liu et al. 2019e], XLNet [Yang et al. 2019], and T5 [Raffel et al. 2020] have led to rapid advancement on the GLUE benchmark, such that within a year of the initial benchmark release, automatic systems were nearing estimates of human performance on the benchmark [Nangia and Bowman 2019].

The release of GLUE also inspired similar multi-task benchmarks. Many of these benchmarks expanded the English-centric (only) language understanding focus of the GLUE benchmark to other languages (such as Mandarin [CLUE, Xu et al. 2020], French [FLUE, Le et al. 2020], Korean [KLUE, Park et al. 2021], Polish [KLEJ (Polish for "glue"), Rybak et al. 2020], and Indonesian [IndoNLU, Wilie et al. 2020]) or crosslingual language understanding (such as XTREME [Hu et al. 2020] or XGLUE [Liang et al. 2020]). Other works extended the multi-task benchmark framework to different settings, such as multimodal language understanding tasks [VALUE, Li et al. 2021] or language generation tasks [GLGE, Liu et al. 2021]. The common methodology of these benchmarks is in assessing the generalizability of NLU models within a particular class of problems or languages by evaluating models across a variety of tasks within the benchmark scope.

In light of the relatively fast progress on the GLUE benchmark due to the proliferation of large pretrained language models, we also set out to create a more challenging general-purpose language understanding benchmark in the style of GLUE, which we describe in Chapter 3.

3 SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems

3.1 Introduction

Recently there has been notable progress across many natural language processing (NLP) tasks, led by methods such as ELMo [Peters et al. 2018], OpenAI GPT [Radford et al. 2018], and BERT [Devlin et al. 2019]. The unifying theme of these methods is that they couple self-supervised learning from massive unlabelled text corpora with effective adapting of the resulting model to target tasks. The tasks that have proven amenable to this general approach include question answering, textual entailment, and parsing, among many others [Devlin et al. 2019; Kitaev et al. 2019, i.a.].

In this context, the GLUE benchmark [Wang et al. 2019b] has become a prominent evaluation framework for research towards general-purpose language understanding technologies. GLUE is a collection of nine language understanding tasks built on existing public datasets, together with private test data, an evaluation server, a single-number target metric, and an accompanying expert-constructed diagnostic set. GLUE was designed to provide a general-purpose evaluation of language understanding that covers a range of training data volumes, task genres, and task

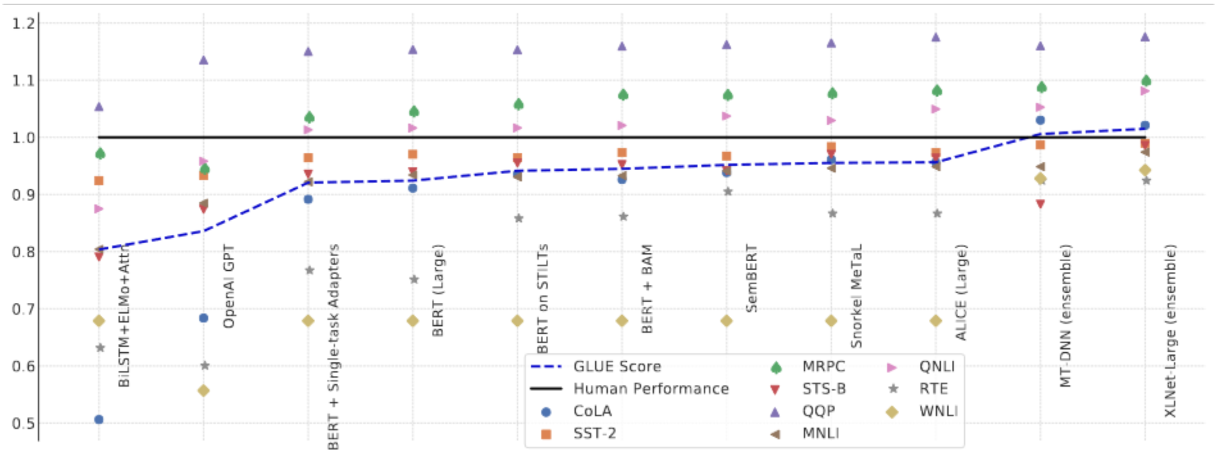


Figure 3.1: GLUE benchmark performance for submitted systems, rescaled to set human performance to 1.0, shown as a single number score, and broken down into the nine constituent task performances. For tasks with multiple metrics, we use an average of the metrics. More information on the tasks included in GLUE can be found in Wang et al. [2019b] and in Warstadt et al. [2019, CoLA], Socher et al. [2013, SST-2], Dolan and Brockett [2005, MRPC], Cer et al. [2017, STS-B], and Williams et al. [2018, MNLI], and Rajpurkar et al. [2016, the original data source for QNLI].

formulations. We believe it was these aspects that made GLUE particularly appropriate for exhibiting the transfer-learning potential of approaches like OpenAI GPT and BERT.

The progress of the last twelve months has eroded headroom on the GLUE benchmark dramatically. While some tasks (Figure 3.1) and some linguistic phenomena (Figure A.2 in Appendix A.2.2) measured in GLUE remain difficult, the current state of the art GLUE Score as of early July 2019 [88.4 from Yang et al. 2019] surpasses human performance [87.1 from Nangia and Bowman 2019] by 1.3 points, and in fact exceeds this human performance estimate on four tasks. Consequently, while there remains substantial scope for improvement towards GLUE's high-level goals, the original version of the benchmark is no longer a suitable metric for quantifying such progress.

In response, we introduce SuperGLUE, a new benchmark designed to pose a more rigorous test of language understanding. SuperGLUE has the same high-level motivation as GLUE: to provide a simple, hard-to-game measure of progress toward general-purpose language understanding technologies for English. We anticipate that significant progress on SuperGLUE should require substantive innovations in a number of core areas of machine learning, including sample-efficient, transfer, multitask, and unsupervised or self-supervised learning.

SuperGLUE follows the basic design of GLUE: It consists of a public leaderboard built around eight language understanding tasks, drawing on existing data, accompanied by a single-number performance metric, and an analysis toolkit. However, it improves upon GLUE in several ways:

More challenging tasks: SuperGLUE retains the two hardest tasks in GLUE. The remaining tasks were identified from those submitted to an open call for task proposals and were selected based on difficulty for current NLP approaches.

More diverse task formats: The task formats in GLUE are limited to sentence- and sentence-pair classification. We expand the set of task formats in SuperGLUE to include coreference resolution and question answering (QA).

Comprehensive human baselines: We include human performance estimates for all benchmark tasks, which verify that substantial headroom exists between a strong BERT-based baseline and human performance.

Improved code support: SuperGLUE is distributed with a new, modular toolkit for work on pretraining, multi-task learning, and transfer learning in NLP, built around standard tools including PyTorch [Paszke et al. 2017] and AllenNLP [Gardner et al. 2017].

Refined usage rules: The conditions for inclusion on the SuperGLUE leaderboard have been revamped to ensure fair competition, an informative leaderboard, and full credit assignment to data and task creators.

The SuperGLUE leaderboard, data, and software tools are available at `super.gluebenchmark.com`.

3.2 Related Work

Much work prior to GLUE demonstrated that training neural models with large amounts of available supervision can produce representations that effectively transfer to a broad range of NLP

Table 3.1: Development set examples from the tasks in SuperGLUE. **Bold** text represents part of the example format for each task. Text in *italics* is part of the model input. *Underlined* text is specially marked in the input. Text in a `monospaced font` represents the expected model output.

BoolQ	**Passage:** *Barq's – Barq's is an American soft drink. Its brand of root beer is notable for having caffeine. Barq's, created by Edward Barq and bottled since the turn of the 20th century, is owned by the Barq family but bottled by the Coca-Cola Company. It was known as Barq's Famous Olde Tyme Root Beer until 2012.* **Question:** *is barq's root beer a pepsi product* **Answer:** `No`
CB	**Text:** *B: And yet, uh, I we-, I hope to see employer based, you know, helping out. You know, child, uh, care centers at the place of employment and things like that, that will help out. A: Uh-huh. B: What do you think, do you think we are, setting a trend?* **Hypothesis:** *they are setting a trend* **Entailment:** `Unknown`
COPA	**Premise:** *My body cast a shadow over the grass.* **Question:** *What's the CAUSE for this?* **Alternative 1:** *The sun was rising.* **Alternative 2:** *The grass was cut.* **Correct Alternative:** `1`
MultiRC	**Paragraph:** *Susan wanted to have a birthday party. She called all of her friends. She has five friends. Her mom said that Susan can invite them all to the party. Her first friend could not go to the party because she was sick. Her second friend was going out of town. Her third friend was not so sure if her parents would let her. The fourth friend said maybe. The fifth friend could go to the party for sure. Susan was a little sad. On the day of the party, all five friends showed up. Each friend had a present for Susan. Susan was happy and sent each friend a thank you card the next week* **Question:** *Did Susan's sick friend recover?* **Candidate answers:** *Yes, she recovered* (`T`), *No* (`F`), *Yes* (`T`), *No, she didn't recover* (`F`), *Yes, she was at Susan's party* (`T`)
ReCoRD	**Paragraph:** *(CNN) Puerto Rico on Sunday overwhelmingly voted for statehood. But Congress, the only body that can approve new states, will ultimately decide whether the status of the US commonwealth changes. Ninety-seven percent of the votes in the nonbinding referendum favored statehood, an increase over the results of a 2012 referendum, official results from the State Electorcal Commission show. It was the fifth such vote on statehood. "Today, we the people of Puerto Rico are sending a strong and clear message to the US Congress ... and to the world ... claiming our equal rights as American citizens, Puerto Rico Gov. Ricardo Rossello said in a news release. @highlight Puerto Rico voted Sunday in favor of US statehood* **Query** For one, they can truthfully say, "Don't blame me, I didn't vote for them, " when discussing the <placeholder> presidency **Correct Entities:** `US`
RTE	**Text:** *Dana Reeve, the widow of the actor Christopher Reeve, has died of lung cancer at age 44, according to the Christopher Reeve Foundation.* **Hypothesis:** *Christopher Reeve had an accident.* **Entailment:** `False`
WiC	**Context 1:** *Room and board.* **Context 2:** *He nailed boards across the windows.* **Sense match:** `False`
WSC	**Text:** *Mark told Pete many lies about himself, which Pete included in his book. He should have been more truthful.* **Coreference:** `False`

tasks [Collobert and Weston 2008; Dai and Le 2015; Kiros et al. 2015; Hill et al. 2016; Conneau and Kiela 2018; McCann et al. 2017; Peters et al. 2018]. GLUE was presented as a formal challenge affording straightforward comparison between such task-agnostic transfer learning techniques. Other similarly-motivated benchmarks include SentEval [Conneau and Kiela 2018], which specifically evaluates fixed-size sentence embeddings, and DecaNLP [McCann et al. 2018], which recasts a set of target tasks into a general question-answering format and prohibits task-specific parameters. In contrast, GLUE provides a lightweight classification API and no restrictions on model architecture or parameter sharing, which seems to have been well-suited to recent work in this area.

Since its release, GLUE has been used as a testbed and showcase by the developers of several influential models, including GPT [Radford et al. 2018] and BERT [Devlin et al. 2019]. As shown in Figure 3.1, progress on GLUE since its release has been striking. On GLUE, GPT and BERT achieved scores of 72.8 and 80.2 respectively, relative to 66.5 for an ELMo-based model [Peters et al. 2018] and 63.7 for the strongest baseline with no multitask learning or pretraining above the word level. Recent models [Liu et al. 2019d; Yang et al. 2019] have clearly surpassed estimates of non-expert human performance on GLUE [Nangia and Bowman 2019]. The success of these models on GLUE has been driven by ever-increasing model capacity, compute power, and data quantity, as well as innovations in model expressivity (from recurrent to bidirectional recurrent to multi-headed transformer encoders) and degree of contextualization (from learning representation of words in isolation to using uni-directional contexts and ultimately to leveraging bidirectional contexts).

In parallel to work scaling up pretrained models, several studies have focused on complementary methods for augmenting performance of pretrained models. Phang et al. [2018] show that BERT can be improved using two-stage pretraining, i.e., fine-tuning the pretrained model on an intermediate data-rich supervised task before fine-tuning it again on a data-poor target task. Liu et al. [2019d,c] and Bach et al. [2018] get further improvements respectively via multi-task finetuning and using massive amounts of weak supervision. Clark et al. [2019b] demonstrate that knowledge distillation [Hinton et al. 2015; Furlanello et al. 2018] can lead to student networks that outperform their teachers. Overall, the quantity and quality of research contributions aimed at the challenges posed by GLUE underline the utility of this style of benchmark for machine learning researchers looking to evaluate new application-agnostic methods on language understanding.

Limits to current approaches are also apparent via the GLUE suite. Performance on the GLUE

Table 3.2: The tasks included in SuperGLUE. *WSD* stands for word sense disambiguation, *NLI* is natural language inference, *coref.* is coreference resolution, and *QA* is question answering. For MultiRC, we list the number of total answers for 456/83/166 train/dev/test questions.

Corpus	|Train|	|Dev|	|Test|	Task	Metrics	Text Sources
BoolQ	9427	3270	3245	QA	acc.	Google queries, Wikipedia
CB	250	57	250	NLI	acc./F1	various
COPA	400	100	500	QA	acc.	blogs, photography encyclopedia
MultiRC	5100	953	1800	QA	$\mathrm{F1}_a$/EM	various
ReCoRD	101k	10k	10k	QA	F1/EM	news (CNN, Daily Mail)
RTE	2500	278	300	NLI	acc.	news, Wikipedia
WiC	6000	638	1400	WSD	acc.	WordNet, VerbNet, Wiktionary
WSC	554	104	146	coref.	acc.	fiction books

diagnostic entailment dataset, at 0.42 R_3, falls far below the average human performance of 0.80 R_3 reported in the original GLUE publication, with models performing near, or even below, chance on some linguistic phenomena (Figure A.2, Appendix A.2.2). While some initially difficult categories saw gains from advances on GLUE (e.g., double negation), others remain hard (restrictivity) or even adversarial (disjunction, downward monotonicity). This suggests that even as unsupervised pretraining produces ever-better statistical summaries of text, it remains difficult to extract many details crucial to semantics without the right kind of supervision. Much recent work has made similar observations about the limitations of existing pretrained models [Jia and Liang 2017; Naik et al. 2018; McCoy and Linzen 2019; McCoy et al. 2019; Liu et al. 2019a,b].

3.3 SuperGLUE Overview

3.3.1 Design Process

The goal of SuperGLUE is to provide a simple, robust evaluation metric of any method capable of being applied to a broad range of language understanding tasks. To that end, in designing SuperGLUE, we identify the following desiderata of tasks in the benchmark:

Task substance: Tasks should test a system's ability to understand and reason about texts in English.

Task difficulty: Tasks should be beyond the scope of current state-of-the-art systems, but solvable by most college-educated English speakers. We exclude tasks that require domain-specific knowledge, e.g. medical notes or scientific papers.

Evaluability: Tasks must have an automatic performance metric that corresponds well to human judgments of output quality. Some text generation tasks fail to meet this criteria due to issues with automatic metrics like ROUGE and BLEU [Callison-Burch et al. 2006; Liu et al. 2016, i.a.].

Public data: We require that tasks have *existing* public training data in order to minimize the risks involved in newly-created datasets. We also prefer tasks for which we have access to (or could create) a test set with private labels.

Task format: We prefer tasks that had relatively simple input and output formats, to avoid incentivizing the users of the benchmark to create complex task-specific model architectures. Still, while GLUE is restricted to tasks involving single sentence or sentence pair inputs, for SuperGLUE we expand the scope to consider tasks with longer inputs. This yields a set of tasks that requires understanding individual tokens in context, complete sentences, inter-sentence relations, and entire paragraphs.

License: Task data must be available under licences that allow use and redistribution for research purposes.

To identify possible tasks for SuperGLUE, we disseminated a public call for task proposals to the NLP community, and received approximately 30 proposals. We filtered these proposals according to our criteria. Many proposals were not suitable due to licensing issues, complex formats, and insufficient headroom; we provide examples of such tasks in Appendix A.2.4. For each of the remaining tasks, we ran a BERT-based baseline and a human baseline, and filtered out tasks which were either too challenging for humans without extensive training or too easy for our machine baselines.

3.3.2 Selected Tasks

Following this process, we arrived at eight tasks to use in SuperGLUE. See Tables 3.2 and 3.1 for details and specific examples of each task.

BoolQ [Boolean Questions, Clark et al. 2019a] is a QA task where each example consists of a short passage and a yes/no question about the passage. The questions are provided anonymously and unsolicited by users of the Google search engine, and afterwards paired with a paragraph from a Wikipedia article containing the answer. Following the original work, we evaluate with accuracy.

CB [CommitmentBank, de Marneffe et al. 2019] is a corpus of short texts in which at least one sentence contains an embedded clause. Each of these embedded clauses is annotated with the degree to which it appears the person who wrote the text is *committed* to the truth of the clause. The resulting task framed as three-class textual entailment on examples that are drawn from the Wall Street Journal, fiction from the British National Corpus, and Switchboard. Each example consists of a premise containing an embedded clause and the corresponding hypothesis is the extraction of that clause. We use a subset of the data that had inter-annotator agreement above 80%. The data is imbalanced (relatively fewer *neutral* examples), so we evaluate using accuracy and F1, where for multi-class F1 we compute the unweighted average of the F1 per class.

COPA [Choice of Plausible Alternatives, Roemmele et al. 2011] is a causal reasoning task in which a system is given a premise sentence and must determine either the cause or effect of the premise from two possible choices. All examples are handcrafted and focus on topics from blogs and a photography-related encyclopedia. Following the original work, we evaluate using accuracy.

MultiRC [Multi-Sentence Reading Comprehension, Khashabi et al. 2018] is a QA task where each example consists of a context paragraph, a question about that paragraph, and a list of possible answers. The system must predict which answers are true and which are false. While many QA tasks exist, we use MultiRC because of a number of desirable properties: (i) each question can have multiple possible correct answers, so each question-answer pair must be evaluated independent of other pairs, (ii) the questions are designed such that answering each question requires drawing facts from multiple context sentences, and (iii) the question-answer pair format more closely matches the API of other tasks in SuperGLUE than the more popular span-extractive QA format does. The paragraphs are drawn from seven domains including news, fiction, and historical text. The evaluation metrics are F1 over all answer-options ($F1_a$) and exact match of each question's set of answers (EM).

ReCoRD [Reading Comprehension with Commonsense Reasoning Dataset, Zhang et al. 2018] is a multiple-choice QA task. Each example consists of a news article and a Cloze-style question about the article in which one entity is masked out. The system must predict the masked out entity from a list of possible entities in the provided passage, where the same entity may be expressed with multiple different surface forms, which are all considered correct. Articles are from CNN and Daily Mail. We evaluate with max (over all mentions) token-level F1 and exact match (EM).

RTE (Recognizing Textual Entailment) datasets come from a series of annual competitions on textual entailment. RTE is included in GLUE, and we use the same data and format as GLUE: We merge data from RTE1 [Dagan et al. 2006], RTE2 [Bar Haim et al. 2006], RTE3 [Giampiccolo et al. 2007], and RTE5 [Bentivogli et al. 2009]. All datasets are combined and converted to two-class classification: *entailment* and *not_entailment*. Of all the GLUE tasks, RTE is among those that benefits from transfer learning the most, with performance jumping from near random-chance (~56%) at the time of GLUE's launch to 86.3% accuracy [Liu et al. 2019d; Yang et al. 2019] at the time of writing. Given the nearly eight point gap with respect to human performance, however, the task is not yet solved by machines, and we expect the remaining gap to be difficult to close.

WiC [Word-in-Context, Pilehvar and Camacho-Collados 2019] is a word sense disambiguation task cast as binary classification of sentence pairs. Given two text snippets and a polysemous word that appears in both sentences, the task is to determine whether the word is used with the same sense in both sentences. Sentences are drawn from WordNet [Miller 1995], VerbNet [Schuler 2005], and Wiktionary. We follow the original work and evaluate using accuracy.

WSC [Winograd Schema Challenge, Levesque et al. 2012] is a coreference resolution task in which examples consist of a sentence with a pronoun and a list of noun phrases from the sentence. The system must determine the correct referrent of the pronoun from among the provided choices. Winograd schemas are designed to require everyday knowledge and commonsense reasoning to solve.

GLUE includes a version of WSC recast as NLI, known as WNLI. Until very recently, no substantial progress had been made on WNLI, with many submissions opting to submit majority class predictions.[1] In the past few months, several works [Kocijan et al. 2019; Liu et al. 2019d] have made rapid progress via a hueristic data augmentation scheme, raising machine performance to 90.4% accuracy. Given estimated human performance of ~96%, there is still a gap between machine and human performance, which we expect will be relatively difficult to close. We therefore include a version of WSC cast as binary classification, where each example consists of a sentence with a marked pronoun and noun, and the task is to determine if the pronoun refers to that noun. The training and validation examples are drawn from the original WSC data [Levesque et al. 2012], as well as those distributed by the affiliated organization *Commonsense Reasoning*.[2] The test examples are derived from fiction books and have been shared with us by the authors of the original dataset. We evaluate using accuracy.

[1] WNLI is especially difficult due to an adversarial train/dev split: Premise sentences that appear in the training set often appear in the development set with a different hypothesis and a flipped label. If a system memorizes the training set, which was easy due to the small size of the training set, it could perform far *below* chance on the development set. We remove this adversarial design in our version of WSC by ensuring that no sentences are shared between the training, validation, and test sets.

[2] http://commonsensereasoning.org/disambiguation.html

3.3.3 Scoring

As with GLUE, we seek to give a sense of aggregate system performance over all tasks by averaging scores of all tasks. Lacking a fair criterion with which to weight the contributions of each task to the overall score, we opt for the simple approach of weighing each task equally, and for tasks with multiple metrics, first averaging those metrics to get a task score.

3.3.4 Tools for Model Analysis

Analyzing Linguistic and World Knowledge in Models GLUE includes an expert-constructed, diagnostic dataset that automatically tests models for a broad range of linguistic, commonsense, and world knowledge. Each example in this broad-coverage diagnostic is a sentence pair labeled with a three-way entailment relation (*entailment*, *neutral*, or *contradiction*) and tagged with labels that indicate the phenomena that characterize the relationship between the two sentences. Submissions to the GLUE leaderboard are required to include predictions from the submission's MultiNLI classifier on the diagnostic dataset, and analyses of the results were shown alongside the main leaderboard. Since this diagnostic task has proved difficult for top models, we retain it in SuperGLUE. However, since MultiNLI is not part of SuperGLUE, we collapse *contradiction* and *neutral* into a single *not_entailment* label, and request that submissions include predictions on the resulting set from the model used for the *RTE* task. We estimate human performance following the same procedure we use for the benchmark tasks (Section A.2.3). We estimate an accuracy of 88% and a Matthew's correlation coefficient (MCC, the two-class variant of the R_3 metric used in GLUE) of 0.77.

Analyzing Gender Bias in Models Recent work has identified the presence and amplification of many social biases in data-driven machine learning models [Lu et al. 2020; Zhao et al. 2018, i.a.]. To promote the detection of such biases, we include Winogender [Rudinger et al. 2018] as an additional diagnostic dataset. Winogender is designed to measure gender bias in coreference resolution systems. We use the Diverse Natural Language Inference Collection [Poliak et al. 2018a] version that casts Winogender as a textual entailment task. Each example consists of a premise sentence with a male or female pronoun and a hypothesis giving a possible antecedent of the pronoun. Examples occur in *minimal pairs*, where the only difference between an example and its pair is the gender of the pronoun in the premise. Performance on Winogender is measured with accuracy and the *gender parity score*: the percentage of minimal pairs for which the predictions are the same. A system can trivially obtain a perfect gender parity score by guessing the same class for all examples, so a high gender parity score is meaningless unless accompanied by high accuracy. We collect non-expert annotations to estimate human performance, and observe an accuracy of 99.7% and a gender parity score of 0.99.

Like any diagnostic, Winogender has limitations. It offers only positive predictive value: A poor bias score is clear evidence that a model exhibits gender bias, but a good score does not mean that the model is unbiased. More specifically, in the DNC version of the task, a low gender parity score means that a model's prediction of textual entailment can be changed with a change in pronouns, all else equal. It is plausible that there are forms of bias that are relevant to target tasks of interest, but that do not surface in this setting [Gonen and Goldberg 2019]. Also, Winogender does not cover all forms of social bias, or even all forms of gender. For instance, the version of the data used here offers no coverage of gender-neutral *they* or non-binary pronouns. Despite these limitations, we believe that Winogender's inclusion is worthwhile in providing a coarse sense of how social biases evolve with model performance and for keeping attention on the social ramifications of NLP models.

3.4 Using SuperGLUE

Software Tools To facilitate using SuperGLUE, we release `jiant` [Wang et al. 2019c],[3] a modular software toolkit, built with PyTorch [Paszke et al. 2017], components from AllenNLP [Gardner et al. 2017], and the `transformers` package.[4] `jiant` implements our baselines and supports the evaluation of custom models and training methods on the benchmark tasks. The toolkit includes support for existing popular pretrained models such as OpenAI GPT and BERT, as well as support for multistage and multitask learning of the kind seen in the strongest models on GLUE.

Eligibility Any system or method that can produce predictions for the SuperGLUE tasks is eligible for submission to the leaderboard, subject to the data-use and submission frequency policies stated immediately below. There are no restrictions on the type of methods that may be used, and there is no requirement that any form of parameter sharing or shared initialization be used across the tasks in the benchmark. To limit overfitting to the private test data, users are limited to a maximum of two submissions per day and six submissions per month.

Data Data for the tasks are available for download through the SuperGLUE site and through a download script included with the software toolkit. Each task comes with a standardized training set, development set, and *unlabeled* test set. Submitted systems may use any public or private data when developing their systems, with a few exceptions: Systems may only use the SuperGLUE-distributed versions of the task datasets, as these use different train/validation/test splits from other public versions in some cases. Systems also may not use the unlabeled test data for the tasks in system development in any way, may not use the structured source data that was used to collect the WiC labels (sense-annotated example sentences from WordNet, VerbNet, and Wiktionary) in any way, and may not build systems that share information across separate *test* examples in any

[3] https://github.com/nyu-mll/jiant

[4] https://github.com/huggingface/transformers

way.

To ensure reasonable credit assignment, because we build very directly on prior work, we ask the authors of submitted systems to directly name and cite the specific datasets that they use, *including the benchmark datasets*. We will enforce this as a requirement for papers to be listed on the leaderboard.

3.5 Experiments

3.5.1 Baselines

Table 3.3: Baseline performance on the SuperGLUE test sets and diagnostics. For CB we report accuracy and macro-average F1. For MultiRC we report F1 on all answer-options and exact match of each question's set of correct answers. AX$_b$ is the broad-coverage diagnostic task, scored using Matthews' correlation (MCC). AX$_g$ is the Winogender diagnostic, scored using accuracy and the gender parity score (GPS). All values are scaled by 100. The *Avg* column is the overall benchmark score on non-AX$_*$ tasks. The bolded numbers reflect the best machine performance on task. *MultiRC has multiple test sets released on a staggered schedule, and these results evaluate on an installation of the test set that is a subset of ours.

Model	**Avg**	**BoolQ**	**CB**	**COPA**	**MultiRC**	**ReCoRD**	**RTE**	**WiC**	**WSC**	**AX$_b$**	**AX$_g$**
Metrics		**Acc.**	**F1/Acc.**	**Acc.**	**F1$_a$/EM**	**F1/EM**	**Acc.**	**Acc.**	**Acc.**	**MCC**	**GPS Acc.**
Most Frequent	47.1	62.3	21.7/48.4	50.0	61.1 / 0.3	33.4/32.5	50.3	50.0	65.1	0.0	100.0/ 50.0
CBoW	44.3	62.1	49.0/71.2	51.6	0.0 / 0.4	14.0/13.6	49.7	53.0	65.1	-0.4	100.0/ 50.0
BERT	69.0	77.4	75.7/83.6	70.6	70.0 / 24.0	72.0/71.3	71.6	**69.5**	**64.3**	23.0	97.8 / 51.7
BERT++	**71.5**	79.0	**84.7**/**90.4**	73.8	70.0 / 24.1	72.0/71.3	79.0	**69.5**	**64.3**	38.0	99.4 / 51.4
Outside Best	-	**80.4**	- / -	**84.4**	**70.4***/**24.5***	**74.8**/**73.0**	**82.7**	-	-	-	- / -
Human (est.)	89.8	89.0	95.8/98.9	100.0	81.8*/51.9*	91.7/91.3	93.6	80.0	100.0	77.0	99.3 / 99.7

BERT Our main baselines are built around BERT, variants of which are among the most successful approach on GLUE at the time of writing. Specifically, we use the `bert-large-cased` variant. Following the practice recommended in Devlin et al. [2019], for each task, we use the simplest possible architecture on top of BERT. We fine-tune a copy of the pretrained BERT model separately for each task, and leave the development of multi-task learning models to future work. For training, we use the procedure specified in Devlin et al. [2019]: We use Adam [Kingma and Ba 2014] with an initial learning rate of 10^{-5} and fine-tune for a maximum of 10 epochs.

For classification tasks with sentence-pair inputs (BoolQ, CB, RTE, WiC), we concatenate the sentences with a [SEP] token, feed the fused input to BERT, and use a logistic regression classifier that sees the representation corresponding to [CLS]. For WiC, we also concatenate the representation of the marked word. For COPA, MultiRC, and ReCoRD, for each answer choice, we similarly concatenate the context with that answer choice and feed the resulting sequence into BERT to produce an answer representation. For COPA, we project these representations into a scalar, and take as the answer the choice with the highest associated scalar. For MultiRC, because each question can have more than one correct answer, we feed each answer representation into a logistic regression classifier. For ReCoRD, we also evaluate the probability of each candidate independent of other candidates, and take the most likely candidate as the model's prediction. For WSC, which is a span-based task, we use a model inspired by Tenney et al. [2019]. Given the BERT representation for each word in the original sentence, we get span representations of the pronoun and noun phrase via a self-attention span-pooling operator [Lee et al. 2017], before feeding it into a logistic regression classifier.

BERT++ We also report results using BERT with additional training on related datasets before fine-tuning on the benchmark tasks, following the STILTs style of transfer learning [Phang et al. 2018]. Given the productive use of MultiNLI in pretraining and intermediate fine-tuning of pretrained language models [Conneau et al. 2017; Phang et al. 2018, i.a.], for CB, RTE, and BoolQ, we use MultiNLI as a transfer task by first using the above procedure on MultiNLI. Similarly, given the similarity of COPA to SWAG [Zellers et al. 2018], we first fine-tune BERT on SWAG. These results are reported as BERT++. For all other tasks, we reuse the results of BERT fine-tuned on just that task.

Other Baselines We include a baseline where for each task we simply predict the majority class,[5] as well as a bag-of-words baseline where each input is represented as an average of its

[5] For ReCoRD, we predict the entity that has the highest F1 with the other entity options.

tokens' GloVe word vectors [the 300D/840B release from Pennington et al. 2014]. Finally, we list the best known result on each task as of May 2019, except on tasks which we recast (WSC), resplit (CB), or achieve the best known result (WiC). The outside results for COPA, MultiRC, and RTE are from Sap et al. [2019], Trivedi et al. [2019], and Liu et al. [2019d] respectively.

Human Performance Pilehvar and Camacho-Collados [2019], Khashabi et al. [2018], Nangia and Bowman [2019], and Zhang et al. [2018] respectively provide estimates for human performance on WiC, MultiRC, RTE, and ReCoRD. For the remaining tasks, including the diagnostic set, we estimate human performance by hiring crowdworker annotators through Amazon's Mechanical Turk platform to reannotate a sample of each test set. We follow a two step procedure where a crowd worker completes a short training phase before proceeding to the annotation phase, modeled after the method used by Nangia and Bowman [2019] for GLUE. See Appendix A.2.3 for details.

3.5.2 Results

Table 3.3 shows results for all baselines. The most frequent class and CBOW baselines do not perform well overall, achieving near chance performance for several of the tasks. Using BERT increases the average SuperGLUE score by 25 points, attaining significant gains on all of the benchmark tasks, particularly MultiRC, ReCoRD, and RTE. On WSC, BERT actually performs worse than the simple baselines, likely due to the small size of the dataset and the lack of data augmentation. Using MultiNLI as an additional source of supervision for BoolQ, CB, and RTE leads to a 2-5 point improvement on all tasks. Using SWAG as a transfer task for COPA sees an 8 point improvement.

Our best baselines still lag substantially behind human performance. On average, there is a nearly 20 point gap between BERT++ and human performance. The largest gap is on WSC, with a 35 point difference between the best model and human performance. The smallest margins are on

BoolQ, CB, RTE, and WiC, with gaps of around 10 points on each of these. We believe these gaps will be challenging to close: On WSC and COPA, human performance is perfect. On three other tasks, it is in the mid-to-high 90s. On the diagnostics, all models continue to lag significantly behind humans. Though all models obtain near perfect gender parity scores on Winogender, this is due to the fact that they are obtaining accuracy near that of random guessing.

3.6 Conclusion

We present SuperGLUE, a new benchmark for evaluating general-purpose language understanding systems. SuperGLUE updates the GLUE benchmark by identifying a new set of challenging NLU tasks, as measured by the difference between human and machine baselines. The set of eight tasks in our benchmark emphasizes diverse task formats and low-data training data tasks, with nearly half the tasks having fewer than 1k examples and all but one of the tasks having fewer than 10k examples.

We evaluate BERT-based baselines and find that they still lag behind humans by nearly 20 points. Given the difficulty of SuperGLUE for BERT, we expect that further progress in multi-task, transfer, and unsupervised/self-supervised learning techniques will be necessary to approach human-level performance on the benchmark. Overall, we argue that SuperGLUE offers a rich and challenging testbed for work developing new general-purpose machine learning methods for language understanding.

3.7 Retrospective

In Chapter 3, we introduced the SuperGLUE benchmark, the successor to the GLUE benchmark that updates it with more diverse and more challenging tasks. Tasks are selected from a pool of community-proposed tasks and after verification that the task is easily solved by humans and beyond the capability of state-of-the-art models at the time. At release, SuperGLUE was indeed challenging for NLP models, and it successfully served as one of the primary evaluation methods for prominent large language models such as RoBERTa [Liu et al. 2019e], T5 [Raffel et al. 2020], and ERNIE [Sun et al. 2021]. However, progress on SuperGLUE improved so quickly that within six months of its release, the best models had already matched human crowdworker performance. The steps taken to ensure the difficulty of the benchmark were not successful in outpacing the speed of model improvements, and it seems unlikely that another round of benchmark creation similar to that of SuperGLUE would produce a benchmark that stands up against the rapid pace of model progress. However, there have been several recent innovations in benchmark creation that seem promising in developing sustainable, challenging benchmarks.

The most obvious spiritual successor to the GLUE and SuperGLUE benchmarks is the Beyond the Imitation Game benchmark [BIG-bench, BIG-bench collaboration 2021].[6] Like SuperGLUE, BIG-bench evaluates NLP models by their ability to perform a diverse set of tasks that have been crowdsourced from the NLP research community. Unlike SuperGLUE, BIG-bench focuses on the few-shot setting, where there are typically very few training examples for a task. Additionally, BIG-bench differs from SuperGLUE in that it sets out to measure whether scaling up language models can eventually solve *any text-based task*. To answer this, BIG-bench not only includes traditional NLP tasks, such as machine reasoning or question answering, but also extremely niche tasks, such as solving crosswords puzzles, recognizing ASCII art, or answering questions about cryobiology in Spanish. In order to produce such a diverse range of tasks, BIG-bench uses an open, light-weight reviewing process for community-proposed tasks. All tasks are defined with a consistent format in order to facilitate easy task reviewing and model evaluation, ultimately allowing for a large volume of tasks to be considered. The current version of BIG-bench consists of 209 tasks across dozens of tasks types.

Another prominent successor to SuperGLUE is the Dynabench platform [Kiela et al. 2021],

[6] https://github.com/google/BIG-bench

which benchmarks NLP models by explicitly trying to create examples that are challenging for existing models. Dynabench offers three key innovations over GLUE and SuperGLUE. First, Dynabench focuses on task examples that existing models get wrong, rather than the average example for a task. In order to facilitate the collection of such hard examples, Dynabench relies on a process called *adversarial data collection*, wherein crowdworkers try to write task examples that "break" current models. The examples that workers write are immediately evaluated against the model, and the worker receives feedback as to whether their example was successful in fooling the model. However, the efficacy of adversarially generated data in evaluating models fairly has recently been called into question [Bowman and Dahl 2021; Phang et al. 2021; Li and Michael 2022]. Second, Dynabench uses these adversarial examples to periodically update the test sets for each task, rather than relying on a single static dataset release. Because the benchmark test sets are *dynamic*, it is not as problematic if they become saturated because they will soon be updated with more challenging examples. Third, users of the benchmark submit models, rather than model predictions. This allows submitted models to be run not only on current and previous dataset releases, but also future datasets that have yet to be collected. Additionally, the best models can then be incorporated into the adversarial data collection process to create even harder test sets for the next data release. Finally, because users submit models and all models are run in consistent environments, Dynabench includes fair measures of model inference time and memory consumption. Users can then use these performance measures, as well as other measures such as accuracy or robustness, to rerank models based on how heavily they weigh these factors.

Dynabench is one example of a broader trend of making more difficult benchmarks by focusing on more challenging, constrained, and adversarial settings. An orthogonal line of work focuses on models' ability to learn to perform new tasks with few training examples (few-shot learning); FewGLUE [Schick and Schütze 2021], FewNLU [Zheng et al. 2022], FewCLUE [Xu et al. 2021] have been developed to standardize evaluation in this data-constrained setting. In response to concerns about the energy consumption of large pretrained language models, benchmarks like

HULK [Zhou et al. 2021] have been developed to incentivize computation- or parameter-efficient models In a similar vein, a modified version of SuperGLUE was used as the shared task of the SustaiNLP workshop [Wang and Wolf 2020], where models were evaluated not only based on their predictive performance, but also on the energy consumed to produce a full set of test predictions. Finally, while SuperGLUE focused on sentence- or paragraph-level tasks, the SCROLLS benchmark [Shaham et al. 2022] evaluates models' ability to perform NLU tasks on long-document inputs, such as books and meeting transcripts, where the inputs are thousands to hundreds of thousands of words long.

4 QAGS: Asking and Answering Questions to Evaluate the Factual Consistency of Summaries

4.1 Introduction

Automatic summarization aims to produce summaries that are succinct, coherent, relevant, and — crucially — factually correct. Recent progress in conditional text generation has led to models that can generate fluent, topical summaries [Lewis et al. 2020]. However, model-generated summaries frequently contain factual inconsistencies, limiting their applicability [Kryscinski et al. 2019].

The problem of factual inconsistency is due in part to the lack of automatic evaluation metrics that can detect such errors. Standard metrics for evaluating generated text are predominantly based on counting *n*-grams, which weigh all *n*-grams equally and are insensitive to semantic errors. This inadequacy leaves human evaluation as the primary method for evaluating the factual consistencies, which has been noted to be challenging even for humans [Daume III and Marcu 2005; Kryscinski et al. 2019], in addition to being slow and costly.

We argue that evaluation metrics that are able to capture subtle semantic errors are required to build better models. In this work, we introduce a general framework for evaluating conditional text generation that is designed to detect factual inconsistencies in generated text with respect to some input. Our framework consists of three steps: (1) Given a generated text, a question generation (QG) model generates a set of questions about the text. (2) We then use question answering (QA) models to answer these questions given both the input and the generated text. (3) A quality score is computed based on the similarity of corresponding answers.

This approach leverages recent progress in QA and QG to ask and answer human readable, on-topic questions [Devlin et al. 2019; Song et al. 2019]. It only assumes access to a question answering dataset to train the QG and QA models, and is applicable to any modality where a QA model is available, e.g. text, images, or knowledge graphs.

We use this framework to develop QAGS (Question Answering and Generation for Summarization), a metric for evaluating the factual consistency of abstractive document summaries. Compared to commonly used automatic metrics such as ROUGE [Lin 2004], QAGS shows dramatically higher correlations with human judgements of factuality, for example achieving a Pearson correlation coefficient of 54.52 on the CNN/DailyMail summarization task, compared to 17.72 for ROUGE-2. QAGS also achieves new state-of-the-art results on evaluating the factuality of summaries, outperforming recently proposed NLI models for this task [Kryscinski et al. 2019].

Finally, we analyse the robustness of QAGS through an ablation study. QAGS shows robustness to the quality of the underlying QG and QA models, the domain of the models, and the number of questions asked. Even under the worst ablation settings, QAGS still has stronger correlation with human judgments than other automatic metrics.

Overall, we contribute the following: (1) We introduce QAGS, an automatic model-based evaluation metric for measuring the factual consistency of model-generated text. (2) We collect a new set of human judgments of factual consistency of model-generated summaries for two summarization datasets. We demonstrate that QAGS correlates with these judgments significantly better than other automatic metrics. (3) We show via ablations that QAGS is robust to a number of factors including underlying model quality and domain mismatch. (4) We analyze the questions and answers produced in computing QAGS to illustrate which parts of summaries are inconsistent.

(5) We will release models and code to compute QAGS.

4.2 Background: Automatically Evaluating Machine Generated Text

Standard approaches to evaluating generated text are primarily based on counting n-gram overlap. These methods assume access to one or more reference texts, and score a generated summary based on the precision and recall of all reference n-grams in the generated summary. We briefly describe the most common metrics in this family, and refer readers to Liu et al. [2016] for further discussion.

ROUGE [Lin 2004] was developed specifically for evaluating automatic summarization, and its variants are the *de facto* standard for such. The most common variant is ROUGE-n (typically $n \in \{1, 2\}$), which computes the F1 score for all reference n-grams in the generated summary. ROUGE-L, another commonly used variant, is the length of the longest common subsequence (possibly non-consecutive) between a summary and references.

BLEU [Papineni et al. 2002] is closely related to ROUGE but was developed for machine translation. BLEU computes the precision of the reference n-grams in the generated summary. METEOR [Lavie and Agarwal 2007] extends BLEU by using an alignment between the generated text and a reference, as well as using stemming and synonym replacement for more flexible n-gram matching.

We identify two key deficiencies when using these n-gram based evaluation metrics to detect factual inconsistencies in generated text.

First, these metrics require one or more reference texts to compare against. Obtaining references can be expensive and challenging, and as such many text generation datasets contain only a single reference. This problem is exacerbated with high-entropy generation tasks, such as summarization or dialogue, where there is a very large number of acceptable outputs. In these

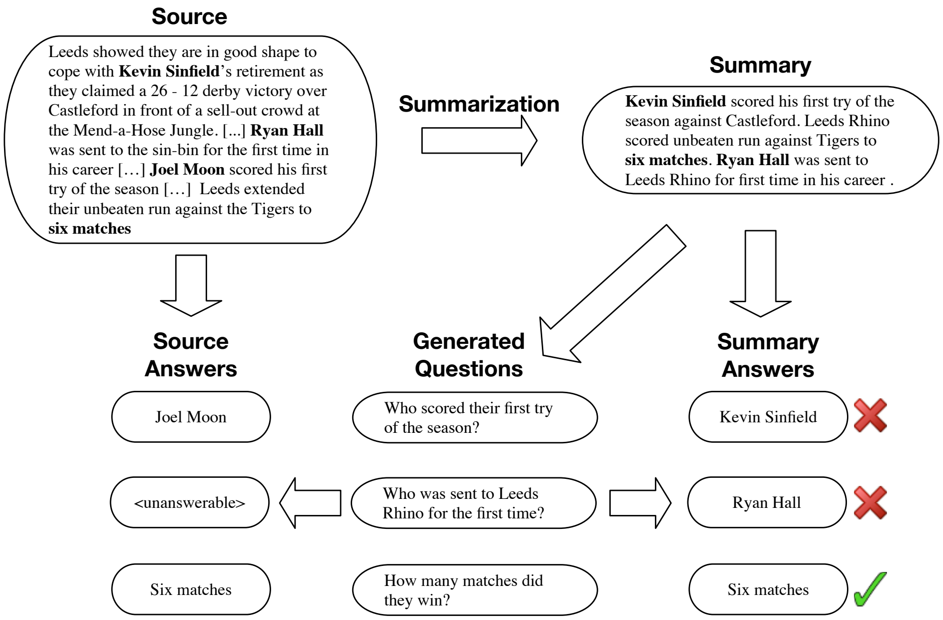

Source
Leeds showed they are in good shape to cope with **Kevin Sinfield**'s retirement as they claimed a 26 - 12 derby victory over Castleford in front of a sell-out crowd at the Mend-a-Hose Jungle. [...] **Ryan Hall** was sent to the sin-bin for the first time in his career […] **Joel Moon** scored his first try of the season […] Leeds extended their unbeaten run against the Tigers to **six matches**
Summarization
Summary
Kevin Sinfield scored his first try of the season against Castleford. Leeds Rhino scored unbeaten run against Tigers to **six matches**. **Ryan Hall** was sent to Leeds Rhino for first time in his career .
Source Answers
Generated Questions
Summary Answers
Joel Moon
Who scored their first try of the season?
Kevin Sinfield
✗
<unanswerable>
Who was sent to Leeds Rhino for the first time?
Ryan Hall
✗
Six matches
How many matches did they win?
Six matches
✓

Figure 4.1: Overview of QAGS. A set of questions is generated based on the summary. The questions are then answered using both the source article and the summary. Corresponding answers are compared using a similarity function and averaged across questions to produce the final QAGS score.

settings, comparing against a single reference is woefully inadequate.

Second, given a reference to compare against, *n*-gram based approach weigh all portions of the text equally, even when only a small fraction of the *n*-grams carry most of the semantic content. Factual inconsistencies caused by minor changes may be drowned out by otherwise high *n*-gram overlap, making these metrics insensitive to these errors. For example, the sentences "I am writing my paper in Vancouver." and "I am not writing my paper in Vancouver." share nearly all unigrams and bigrams despite having the opposite meaning.

4.3 A Framework for Automatically Evaluating Factual Consistency

We introduce a framework for automatically detecting factual inconsistencies in generated text while also addressing the deficiencies of current approaches. Let X and Y be sequences of tokens coming from a vocabulary V where X is a source text and Y is a summary of X. We define $p(Q|Y)$ as a distribution over all possible questions Q given summary Y, and $p(A|Q, X)$ and $p(A|Q, Y)$ as distributions over all possible answers A to a particular question Q given either the source X or the summary Y. We constrain the questions Q and answers A to also be sequences of tokens from V. Then the factual consistency of the summary Y is

$$E_{Q \sim p(Q|Y)} \big[D\big(p(A|Q, X), p(A|Q, Y)\big) \big], \tag{4.1}$$

where D is some function measuring the similarity of the two answer distributions. This expression is maximized when Y contains a subset of the information in X such that it produces the same answer for any question from $p(Q|Y)$. This happens trivially when $Y = X$, i.e. we take X as its own summary, but in many cases this solution is unacceptable.

This framework addresses the two issues with n-gram based approaches. Instead of requiring a reference to compare against, our framework asks questions based on the generation itself, and compares answers with the provided source text. Also, the use of questions focuses the metric on the semantically relevant parts of the generated text, rather than weighting all parts of the text equally.

In practice, exactly computing the expectation in Equation 4.1 is intractable due to the large space of possible questions. One potential workaround is to randomly sample questions from $p(Q|Y)$, but this suffers from high variance and requires many samples to obtain a good estimate. Instead, we focus on producing highly probable questions, e.g. as produced by beam search, which may be biased in the limit, but will require fewer questions to estimate because of the higher quality of the questions.

4.4 QAGS

Using this framework requires specifying the question distribution $p(Q|Y)$, the answer distributions $p(A|Q,*)$, and the answer similarity function D. We apply this framework to summarization to develop QAGS and describe our instantiations of these components.

Question Generation To instantiate $p(Q|Y)$, we draw on recent work on automatic question generation (QG), which models this distribution using neural seq2seq models [Du et al. 2017; Krishna and Iyyer 2019]. We over-sample questions, and then filter out low quality questions as follows.

First, we train and generate from answer-conditional QG models. During training, the model receives both the answer and the source article, and is trained to maximize the likelihood of the paired question. At test time, given a summary Y, we determine candidate answers. We condition on these answers and the summary to generate questions.

Next, we filter out low-quality questions using a number of heuristics, such as duplicates and questions less than three tokens long. We also found it especially useful to run the QA model (see next section) on all of the candidate questions, and filter out questions for which the QA model predicted no answer or a different answer than expected.

Question Answering We instantiate the answer distributions $p(A|Q,*)$ as extractive QA models, for simplicity. In using extractive QA models, we assume the facts are represented as text spans in the article and summary. Future work should explore using abstractive QA models, which could match paraphrases of the same answer.

Answer Similarity We use token-level F1 to compare answers, which is standard for extractive QA and equivalent to defining D as

$$F1(\arg\max p(A|Q,X), \arg\max p(A|Q,Y))$$

The QAGS Score Given these components, we obtain the QAGS score of a generation by (1) generating K questions conditioned on the summary, (2) answering the questions using both the source article and the summary to get two sets of answers, (3) comparing corresponding answers using the answer similarity metric, and (4) averaging the answer similarity metric over all questions. We depict this process in Figure 4.1.

4.5 Experiments

4.5.1 Human Evaluation

We test whether QAGS accurately measures the factual consistency of a summary with respect to a source article by computing correlations with human judgments of factual consistency.

Datasets We focus on abstractive summarization, which is particularly interesting because factual consistency with the original text is crucial to usability, and a lack of such consistency has plagued abstractive neural summarization models [Cao et al. 2018; Falke et al. 2019; Kryscinski et al. 2019, i.a.]. To compare with prior work on evaluating summarization, we use two common abstractive summarization datasets, CNN/Daily Mail [CNNDM, Hermann et al. 2015; Nallapati et al. 2016] and XSUM [Narayan et al. 2018a].

CNN/DM is a standard dataset for summarization that consists of CNN and DailyMail articles. Each reference summary consists of the concatenation of three editor-written, bullet point highlights. For summaries, we use 235 test outputs from Gehrmann et al. [2018].

Metric	CNN/DM	XSUM
ROUGE-1	28.74	13.22
ROUGE-2	17.72	8.95
ROUGE-L	24.09	8.86
METEOR	26.65	10.03
BLEU-1	29.68	11.76
BLEU-2	25.65	11.68
BLEU-3	23.96	8.41
BLEU-4	21.45	5.64
BERTScore	27.63	2.51
QAGS	**54.53**	**17.49**

Table 4.1: Summary-level Pearson correlation coefficients between various automatic metrics and human judgments of correctness for summarization datasets. All correlations are significant at $p < .01$ and $p < .05$ for CNN/DM and XSUM, respectively. QAGS obtains substantially higher correlations than all other automatic metrics.

XSUM was created by taking the first sentence of a news article as the summary, and using the rest of the article as the source. Consequently, XSUM summaries are significantly more abstractive than those of CNN/DM, and extractive summarization models perform poorly on this dataset.

We found that while the XSUM summaries are more abstractive, frequently there are facts (e.g. first names) in the summary that are not available in the "article". This quirk made it especially difficult for humans and QAGS to tell when factual errors were being made by the summarization model. To remedy this, for human evaluation and QAGS, we prepend the summary back to the "article". We use a subset of 239 test outputs from BART fine-tuned on XSUM [Lewis et al. 2020].

Annotation Protocol We collect human judgments on Amazon Mechanical Turk[1] via ParlAI [Miller et al. 2017]. We present summaries one sentence at a time, along with the entire article. For each summary sentence, the annotator makes a binary decision as to whether the sentence is factually consistent with the article. Workers are instructed to mark non-grammatical sentences as not consistent, and copies of article sentences as consistent. Workers are paid $1 per full

[1] https://www.mturk.com/

summary annotated. See Appendix A.3.1 for further details.

We collect 3 annotations per summary. To obtain a single consistency score per summary, we first take the majority vote for each sentence, then average the binary scores across summary sentences to produce a final score.

Inter-annotator agreement as measured by Krippendorff's α is 0.51 and 0.34 for CNN/DM and XSUM, respectively indicating "moderate" and "fair" agreement [Ageeva et al. 2015]. While not perfect, these agreement numbers are in-line with similar figures from previous work on summarization evaluation [Daume III and Marcu 2005].

4.5.2 Experimental Details

Question Generation We train answer-conditional QG models by fine-tuning a pretrained BART language model [Lewis et al. 2020] on NewsQA [Trischler et al. 2017], a dataset consisting of CNN articles and crowdsourced questions. During training, the model receives the concatenation of the source article and an answer, and is trained to predict the question. The answer, source article, and question are concatenated with intervening special tokens to mark the boundaries.

At test time, the model receives the concatentation of a summary and an expected answer, and outputs question candidates. For each summary, we extract 10 named entities and noun phrases as answer candidates using the `en-web-sm` spaCy model.[2] For each summary-answer pair, we generate questions using beam search with width 10, for a total of 100 question candidates. We experimented with generating via top-k [Holtzman et al. 2019] and top-p [Fan et al. 2018] sampling, but the generated questions, while diverse, were noisy and frequently nongrammatical. After filtering, we use the $K = 20$ most probable questions. If a summary has too few filtered questions, we randomly sample questions to reach the required number. For additional filtering and training details, see Appendix A.3.2. We implement these models with `fairseq` [Ott et al. 2019].

[2] https://spacy.io/api/entityrecognizer

Question Answering We train extractive QA models by fine-tuning BERT [Devlin et al. 2019] on SQuAD2.0 [Rajpurkar et al. 2018]. We use the `large-uncased` BERT variant via the `transformers` library [Wolf et al. 2020].

We found that allowing the model to predict that a question is unanswerable, as is the case in SQuAD2.0, is particularly useful in filtering out bad questions, as questions based on hallucinated facts in the summary should be unanswerable using the source article.

Baselines We compare against a number of automatic evaluation metrics: ROUGE [Lin 2004], METEOR [Lavie and Agarwal 2007], BLEU [Papineni et al. 2002], and BERTScore [Zhang et al. 2019a]. The latter uses BERT representations to compute an alignment between generation and reference tokens, and which is then used to compute a soft version of unigram F1. We use the `large-uncased` BERT variant.

4.5.3 Results

We present Pearson correlations between human-judged consistency scores and various automatic metrics in Table 4.1. For CNN/DM, all results are significant with $p < 0.01$; for XSUM, all results are significant with $p < .05$. QAGS strongly outperforms other automatic evaluation metrics in terms of correlation with the summary-level human judgments of factual consistency. BLEU and ROUGE perform comparably, and lower order n-gram metrics work better. BERTScore matches the best n-gram metrics on CNN/DM, but the worst overall on XSUM.

On CNN/DM, QAGS obtains nearly twice the correlation of the next best automatic metric (BLEU-1). We speculate that this large increase is due to the sensitivity of the QA model to the sentence fusing behavior exhibited in many summarization models trained on CNN/DM [Lebanoff et al. 2019]. When two sentences are fused to produce an incorrect summary statement, the QA model produces different answers when using the source article than when using the summary.

On XSUM, all metrics correlate worse with human judgments than on CNN/DM, which re-

QA model	SQuAD (F1)	CNN/DM (Pear.)	XSUM (Pear.)
bert-base	75.95	55.20	20.71
bert-large	81.57	54.53	17.49
bert-large-wwm	84.36	51.36	18.07

Table 4.2: Pearson correlations between human judgments of factual consistency and QAGS using QA models of different qualities, as measured by performance on the SQuAD2.0 development set (F1). The correlations are stable across QA model quality.

NewsQA (ppl.)	CNN/DM (Pear.)	XSUM (Pear.)
5.48	54.53	17.49
9.50	50.09	19.93
18.56	47.92	16.38

Table 4.3: Pearson correlations between human judgments of factual consistency and QAGS with QG models of varying quality, as measured by perplexity on the NewsQA development set. We see some decrease in correlation on CNN/DM as QG perplexity increases, though we do not see a similar trend for XSUM.

flects the fact that XSUM is more abstractive. QAGS still outperforms the next best automatic metric.

4.5.4 Ablations

A potential issue with model-based evaluation is that the quality of the evaluation metric may depend heavily on specific hyperparameter settings. We explore the extent to which this is true with QAGS by performing ablations on several factors.

Model Quality We first consider the degree to which the quality of the underlying models impacts their evaluation capabilities.

For QA quality, we answer this question by training QA models of varying quality by fine-tuning different versions of BERT on SQuAD. We present results in Table 4.2. The QA models perform similarly despite substantially different performances on the SQuAD development set.

# Questions	CNN/DM	XSUM
5	41.61	15.63
10	41.17	15.49
20	54.53	17.49
50	57.94	17.74

Table 4.4: Pearson correlation coefficients between QAGS scores with varying number of questions and human judgments of correctness for summarization datasets. The correlation increases with the number of questions used, but with decreasing marginal benefit.

Surprisingly, using the best QA model (`bert-large-wwm`) does not lead to the best correlations with human judgments. On CNN/DM, `bert-large-wwm` slightly underperforms `bert-base` and `bert-large`. On XSUM, `bert-base` slightly outperforms the other two BERT variants. These results indicate that QAGS is fairly robust to the quality of the underlying QA model, though we note that BERT is a strong QA baseline, and using weaker QA models might lead to larger performance dropoffs.

To ablate QG quality, we use models with increasing perplexity on the NewsQA development set. Results in Table 4.3 show that QAGS is robust to the QG model quality, with some decrease in correlation with human judgments as perplexity increases on CNN/DM, and no clear trend on XSUM. Even the weakest QG model still significantly outperforms all other automatic metrics in Table 4.1.

Domain Effects Our approach relies on having a labeled dataset to train QG and QA models. However, for relatively niche domains, such a labeled QA/QG dataset may not exist. Instead, we may need to resort to using models trained on out-of-domain data, leading to domain shift effects that negatively impact the quality of the QAGS scores. We simulate this setting by fine-tuning the QG model on SQuAD, which is of similar size to NewsQA but drawn from Wikipedia articles rather than CNN articles, which exactly matches the genre of the summarization datasets.

Evaluating with this QG model, we get correlations of 51.53 and 15.28 with human judgments on CNN/DM and XSUM respectively, versus 54.53 and 17.49 when using the NewsQA-tuned QG model. The drop in performance indicates a negative domain shift effect. However using the SQuAD-tuned QG model still substantially outperforms all other automatic metrics, again pointing to the robustness of QAGS.

Number of Questions Next, we investigate the correlation with human judgments when varying the number of questions used. Results in Table 4.4 show that increasing the number of questions used improves correlations with human judgments. We observe a large increase when moving from 10 to 20 questions, and a smaller increase from 20 to 50 questions, indicating decreasing marginal benefit moving beyond 50 questions. However, we observe frequent clusters of generated questions that only differ by a few tokens. Encouraging greater diversity when generating questions might lead to better correlations when more questions are used. Still, with just 5 questions used QAGS substantially outperforms other automatic metrics, indicating its robustness.

Answer Similarity Metric Finally, we consider using exact match as an alternative answer similarity metric. Exact match is another common evaluation metric for extractive QA, and is more restrictive than F1. When using EM, we obtain Pearson correlations with human judgments of 45.97 and 18.10 on CNN/DM and XSUM, as opposed to 54.53 and 17.49 when using F1.

4.6 Re-ranking with QAGS

Several works explore the use of natural language inference (NLI) models to detect factual consistency in generated text [Welleck et al. 2019; Falke et al. 2019]. We compare against these methods by evaluating on the sentence ranking experiment from Falke et al. [2019]. The experiment uses 373 triplets of source sentences from CNN/DM and two summary sentences generated from the model from Chen and Bansal [2018]. One summary sentence is factually consistent with the source sentence, and the other is inconsistent. A metric (or model) is evaluated based on how often it ranks the consistent sentence higher than the inconsistent sentence.

Model/Metric	% Correct (↑)
Random	50.0%
BERT NLI	64.1%
ESIM	67.6%
FactCC	70.0%
QAGS	**72.1%**

Table 4.5: Results on the sentence ranking task from Falke et al. [2019]. Results using BERT NLI and ESIM are from Falke et al. [2019]; FactCC is from Kryściński et al. [2020]. QAGS performs best.

We present the results in Table 4.5. Results using two NLI models fine-tuned on MultiNLI [Williams et al. 2018], BERT NLI, and ESIM [Chen et al. 2017], are from Falke et al. [2019]. FactCC [Kryscinski et al. 2019] is an NLI-based fact-checking model that is trained on a dataset tailor made for detecting factual inconsistencies in generated text. QAGS outperforms these methods, while requiring no special supervision for this task.

4.7 Qualitative Analysis

Interpreting QAGS The questions and answers produced in computing QAGS are directly interpretable, and highlight errors in summaries. We present examples of articles, summaries, and the QAGS questions and answers in Table 4.6.

On the first example (Table 4.6, top), QAGS detects several factual inconsistencies in the generated summary: The summary mistakes the first name of the attacker, the location of the attack, and the weapons used. Because the QG model focuses on these details, QAGS is able to correctly penalize the summary for its hallucinations. Because the answer candidates used are mostly named entities and noun phrases, QAGS is particularly effective at detecting errors of this kind. Using more diverse answer candidates may broaden the set of inconsistencies that QAGS is able to detect.

The second example (Table 4.6, bottom), illustrates failure modes of QAGS. For example, the QA model incorrectly marks question 2 as unanswerable. On question 4, both answers produced

Article: On Friday, 28-year-old Usman Khan stabbed reportedly several people at Fishmongers' Hall in London with a large knife, then fled up London Bridge. Members of the public confronted him; one man sprayed Khan with a fire extinguisher, others struck him with their fists and took his knife, and another, a Polish chef named Łukasz, harried him with a five-foot narwhal tusk. [. . .]

Summary : On Friday afternoon , a man named Faisal Khan entered a Cambridge University building and started attacking people with a knife and a fire extinguisher .

Question 1: What did the attacker have ?

Article answer: a large knife **Summary answer:** a knife and a fire extinguisher

Question 2: When did the attack take place ?

Article answer: Friday **Summary answer:** Friday afternoon

Question 3: What is the attacker's name ?

Article answer: Usman Khan **Summary answer:** Faisal Khan

Question 4: Where did the attack take place ?

Article answer: Fishmongers' Hall **Summary answer:** Cambridge University building

---

Article: In findings published on Wednesday in the journal PLOS ONE, an international team of scientists report ancient Egyptians captured sacred ibises (Threskiornis aethiopicus) from the wild for use in ritual sacrifice rather than domesticating the birds. [. . .] The team collected DNA samples from mummified birds collected from six separate catacombs including sites at Abydos, Saqqara, and Tuna el-Gebel with permission from the Egyptian Ministry of State for Antiquity, and several museums offered to send tissue samples from the mummified ibises in their collections. [. . .]

Summary : Archaeologists have used DNA samples from ancient ibis birds to determine whether the birds were domesticated or sacrificed in ancient Egypt

Question 1: Archaeologists have used what to determine whether the birds were domesticated ?

Article Answer: hatchery structures **Summary Answer**: DNA samples

Question 2: Who used DNA samples to determine whether the birds were domesticated ?

Article Answer: [NO ANSWER] **Summary Answer:** Archaeologists

Question 3: What are archeologists using to determine whether the birds were domesticated ?

Article Answer: DNA samples **Summary Answer:** DNA samples

Question 4: Where were the birds found?

Article Answer: six separate catacombs **Summary Answer:** ancient Egypt

Table 4.6: Example questions and answers generated when computing QAGS. The questions are overwhelmingly fluent and relevant. The answers indicate which tokens in the summary are factually consistent or inconsistent. The news articles are originally from https://en.wikinews.org/wiki/Bystanders_foil_knife-weilding_man_on_London_Bridge_with_fire_extinguisher,_whale_tusk and https://en.wikinews.org/wiki/Ancient_Egyptians_collected_wild_ibis_birds_for_sacrifice,_says_study.

are correct, but because they have no common tokens, they are marked inconsistent by QAGS.

Error Analysis The interpretability of QAGS allows for error analysis on the metric. We manually annotate 400 triplets of generated questions, article answers, and summary answers that are produced in computing QAGS on the XSUM summaries, and label them by the quality of the generated questions, predicted answers, and answer similarity scores.

Among the generated questions, 8.75% are nonsensical, while 3.00% are well-formed but unanswerable using the generated summary they were conditioned upon. These figures indicate that the vast majority of questions are understandable and on-topic. We frequently observe multiple questions with slightly different wordings, which is likely due to the low number of answer candidates in XSUM summaries (which are one sentence long) and due to beam search. 8.25% of questions are well-formed but unanswerable using the source, which is usually due to a hallucinated fact in the summary that the QG model turns into a question.

Among predicted answers, 1.75% of questions are potentially answerable using the summary, but are incorrectly answered. This percentage increases to 32.50% for the article, which indicates that the transfer ability of the QA model is lacking. In a small number of cases, we found that while a question had a single answer in the summary, it could have multiple answers in the article.

Finally, for 8.00% of the examples, the question is answered correctly using both the article and summary, but the answers have high lexical variation such that F1 score fails to detect their similarity. While this happens in a relatively small number of cases, exploring similarity metrics other than *n*-gram based approaches could be useful.

Limitations We emphasize that QAGS and our overall framework are specifically designed to detect factual inconsistencies in generated summaries relative to the source article. QAGS does not measure other desirable properties of generated text, including fluency, readability, or factual recall. We therefore recommend using QAGS in conjunction with complementary evaluation metrics.

The choices of QG and QA models in QAGS are particular to abstractive summarization and may require adaptation to be used for other conditional text generation tasks. For example, we expect that extractive summarization models may obtain nearly perfect QAGS scores because facts and statements are directly copied from the source article.

4.8 Related Work

Automatic summarization and its evaluation are long-standing lines of work in NLP, dating at least as far back as the Document Understanding Conferences [Chali and Kolla 2004]. The primary evaluation metric then and now is ROUGE [Lin 2004], though much work has demonstrated the limited ability of ROUGE and its relatives to evaluate summaries [Dorr et al. 2004; Liu and Liu 2009; Kedzie et al. 2018, i.a.]. Other metrics have focused on specific aspects of summarization quality, including content selection [Nenkova and Passonneau 2004], relevance prediction [Daume III and Marcu 2005], and many more.

The idea of evaluating summaries by their ability to answer a set of questions is also long-standing [Mani et al. 1999]. Like our work, Eyal et al. [2019] and Scialom et al. [2021] extend this line of work by incorporating neural network modules. We diverge from these works in two important ways. First, both works use Cloze-style questions, which are generated by masking entities in either the source document or the reference summary. We instead generate the questions with a model, allowing a much greater range of questions. Second, we produce questions conditioned on the generated summary, rather than the reference summary or source article. Producing questions from the generated summary is more appropriate for verifying the accuracy of the text, whereas using the reference or source measures content selection.

There has been a recent resurgence of work leveraging NLU models for evaluating the factuality of generated text. Goodrich et al. [2019] use information extraction models to measure factual overlap, but facts are restricted to pre-defined schemas. Falke et al. [2019] investigate the use of NLI models to evaluate the factual correctness of CNN/DM summaries, and conclude that current NLI models are too brittle to be reliably used in this manner. Kryscinski et al. [2019] train a NLI-based fact-checking model by building a dataset of factual inconsistencies based on noise heuristics. Our QA approach allows a finer-grained analysis, because NLI operates on complete sentences, whereas QAGS can ask many different questions about the same sentence.

4.9 Conclusion

We introduce a framework for automatically detecting factual inconsistencies in conditionally generated texts and use this framework to develop QAGS, a metric for measuring inconsistencies in abstractive summarization. QAGS correlates with human judgments of factuality significantly better than standard automatic evaluation metrics for summarization, and outperforms related NLI-based approaches to factual consistency checking. QAGS is naturally interpretable: The questions and answers produced in computing QAGS indicate which tokens in a generated summary are inconsistent and why.

The framework we present is general, and extending it to other conditional text generation tasks such as image captioning or machine translation is a promising direction. Inspecting the generated questions and answers, we identify the transfer ability of QA models and the rigidity of F1 score as a measure of answer similarity as two key performance bottlenecks. We expect improvements in either would straightforwardly improve the quality of QAGS evaluation. Additionally, incorporating a content selection mechanism to focus the generated questions on salient facts is a promising direction. Overall, we believe QAGS demonstrates the potential of this framework to quantify and incentivize factually consistent text generation.

4.10 Retrospective

There have been many similar works on detecting hallucinations and measuring faithfulness of generated text that have been published around and after the release of QAGS. FEQA [Durmus et al. 2020], released concurrently with QAGS, proposes a similar pipeline of automatically generating questions about spans in the candidate summary, answering question using both the summary and input document, then comparing corresponding answers. Fabbri et al. [2021] extend QAGS with improved QA components while Scialom et al. [2021] extend it by jointly measuring faithfulness and content selection with QA models.

A competing line of work uses natural language inference models to detect hallucinations in text [Falke et al. 2019; Kryscinski et al. 2019; Laban et al. 2021; Goyal and Durrett 2020]. Whether it is better to use QA or NLI models for detecting hallucinations remains an open question. Several works have tried to compare the various methods, as well as standardize meta-evaluation of evaluation metrics for faithfulness, though there is positive evidence for both classes of models [Maynez et al. 2020; Gabriel et al. 2021; Laban et al. 2021; Fabbri et al. 2021; Pagnoni et al. 2021; Honovich et al. 2022].

The above metrics are part of a broader trend of incorporating pretrained models into automatic evaluation metrics. While the aforementioned metrics focus on evaluating faithfulness and detecting hallucinations, many recent metrics try to predict a general quality score of the candidate text. Like ROUGE and BLEU, BERTScore [Zhang et al. 2019a] computes similarity between a reference and a candidate generation, except similarity is defined as cosine similarity in the BERT embedding space. BLEURT [Sellam et al. 2020a] is a regression model that is trained to directly predict human judgments of translation quality that has achieved good results on machine translation metrics development shared tasks [Sellam et al. 2020b; Pu et al. 2021]. BARTScore [Yuan et al. 2021] defines the goodness of a generation as the token-averaged log-probability of the generation using a BART pretrained model.

In parallel to the rise of metrics for measure generation faithfulness, many works have explored various methods for mitigating hallucinations in model outputs. Given the rise of evaluation metrics for faithfulness, a natural approach is to train models by optimizing these metrics. QUALS [Nan et al. 2021a] does this by training summarization models using a soft, differentiable version of QAGS so that the model receives signal to produce faithful outputs during training. An orthogonal line of work trains controllable text generation models where one controllable features is whether or not the model should generate a hallucination in the output [Filippova 2020; Rashkin et al. 2021; Choubey et al. 2021]. At test time, the model is always set to produce generations without any hallucinations. This line of work typically uses heuristics to extract signal for which examples in the training data contain hallucinations. Another line of work modifies the decoding procedure to only generate faithful tokens, e.g. only generating tokens with high confidence [Tian et al. 2019] or that verified to be supported by the input document [Zhao et al. 2020].

Finally, a number of recent studies have identified that the datasets used to train and evaluate text generation models themselves contain hallucinations or unsupported facts. For example, many dialogue datasets contain utterances that draw on background information not available in from the dialogue history [Rashkin et al. 2021; Dziri et al. 2022]. For summarization, a significant portion of the most popular benchmark datasets, CNN/DailyMail [Nallapati et al. 2016] and XSUM [Narayan et al. 2018b], contain information that is not directly supported by the input article [Kryscinski et al. 2019; Tejaswin et al. 2021; Nan et al. 2021b]. Models that are trained on these datasets will produce similar noise patterns in their generations, which points to dataset quality as a partial source of model hallucinations. Given these the necessity of high-quality datasets for evaluating text generation systems, in Chapter 5, we explore methods for creating a new summarization dataset that is free of these issues.

5 SQuALITY: A Quality Dataset for Question-Focused Summarization

5.1 Introduction

Research on automatic text summarization generally requires adequate benchmark datasets. Existing datasets in this area often have issues that seriously limit their usability: For instance, summaries from the popular scraped benchmark summarization dataset CNN/DailyMail [Nallapati et al. 2016] contain HTML artifacts, links to other news articles, and other types of noise [Kryscinski et al. 2019; Tejaswin et al. 2021].

A common approach to creating summarization datasets is to develop heuristics to extract pseudo-summaries from existing texts. While scraped summaries can be cleaned of noise, these heuristics can lead to more fundamental data artifacts. For example, the XSum dataset [Narayan et al. 2018b] was created by extracting the first sentence of a news article to act as the summary for the rest of the document. However, studies have found that 30–50% of summaries created this way contain facts that are unsupported by the rest of the article [Tejaswin et al. 2021; Nan et al. 2021b]. Models trained on this dataset learn to repeat this noise pattern by hallucinating facts in their outputs. It appears that known heuristics do not produce reliable data.

Another approach to creating summarization datasets relies on serendipity in finding naturally occurring summaries. For example, the arXiv and PubMed datasets [Cohan et al. 2018] use the abstracts of scientific papers as summaries of the papers. BigPatent [Sharma et al. 2019] and GovReport [Huang et al. 2021] use expert-written summaries that come with patent filings and government reports, respectively. While these summaries are likely high-quality, the domain of the data poses a significant challenge for system evaluation: Automatic evaluation metrics for summarization are unreliable [Kryscinski et al. 2019; Gehrmann et al. 2022], but the summaries are too technical and jargonistic for non-specialist human raters to evaluate reliably. Because we rely on chance in finding these summaries, we are beholden to whatever domain they come from, rather than the domain we are interested in.

Relying on finding and scraping summarization data is also problematic in that, often, the found data is proprietary and not freely distributable. For example, many researchers and organizations are unwilling to host or distribute the CNN/DailyMail dataset,[1] despite it being one of the most popular summarization datasets to experiment on. Similarly, several recent summarization datasets built on data such as scientific journal papers [Meng et al. 2021] or SparkNotes book summaries [Ladhak et al. 2020; Kryściński et al. 2021] have never been made available to researchers. The dataset creators instead ask potential users to re-scrape them, which can be a serious obstacle to reproducibility.

In this work, we propose a crowdsourcing protocol for collecting original summaries free of these issues. Crowdsourcing summaries has been under-explored because straightforward approaches for doing so are quite labor-intensive. While our protocol is still fairly demanding, we structure it in a way that makes the cost per summary more tractable (~$6/summary) while also including incentives and checks to ensure the summaries are high-quality. The protocol does not rely on finding naturally occurring summaries and is agnostic to the input documents used, so we are free to choose the input documents we want to summarize. We use short stories from Project Gutenberg to avoid the aforementioned domain and licensing issues.

[1] See discussion here.

We use this protocol to collect SQuALITY[2] (Summary-format QUestion Answering with Long Input Texts, Yes!), a dataset for question-focused abstractive summarization of short stories. SQuALITY summaries are created by having trained writers read short stories, then ask questions about different aspects of the story. The writers then answer the questions by writing summaries focusing on that aspect. Each question is answered by four different annotators, who then review each other's work to ensure the data is high-quality. In total, SQuALITY consists of 100 stories, 500 questions, and 2000 summaries.[3]

Overall, we make the following contributions:

1. We develop a crowdsourcing protocol for collecting summaries that partially ameliorates the high cost of crowdsourcing long textual responses while maintaining data quality.

2. We use this protocol to collect SQuALITY, an abstractive summarization dataset. SQuALITY is question-focused, multi-reference, and distributed with a CC BY license.

3. We conduct preliminary experiments on SQuALITY with pretrained language models using human evaluation. We find that state-of-the-art summarization models produce summaries that are significantly worse than human-written summaries.

4. We identify that common automatic evaluation metrics for summarization correlate very poorly with human judgments of quality. We also find that having multiple references when computing automatic evaluation metrics does not improve the correlation of the metric.

SQuALITY is a challenging benchmark for long-context text generation models. We will make the SQuALITY dataset, our baseline models, and our templates for human evaluation of models publicly available upon publication.

[2] Named because it uses many of the same stories as the multiple choice QA dataset QuALITY [Pang et al. 2021b].

[3] This paper releases SQuALITY v1.0. We will soon release SQuALITY v1.1, which consists of 127 stories.

Title: Pick A Crime (https://www.gutenberg.org/ebooks/51656)	
Q: What is the CPA and what does it do?	
The Crime Prevention Association is an organization that stops crime. Instead of capturing criminals, the goal of the Association is to prevent the crime from ever happening. They implement thousands of crime-prevention methods and devices. There are many amateur cops who constantly follow criminals around in hopes of catching them in the act so that they may be hailed a hero and...	The CPA is Crime Prevention Organization. It fights crime by all means and reduces its rates to a very small level. They put microphones and detectors everywhere to hear the conspiracies. They place robots as bartenders to control the level of alcohol in visitors to prevent them being drunk. They make all the women learn self-defense. The organization's made crime almost impossible...
The CPA, Crime Prevention Association, is a system that detects different kinds of crimes and prevents them from happening. Thousands of robots and devices make crimes impossible. The association will not punish any crime, instead, the criminal will be send to a CPA hospital for some treatments that will result in getting the best jobs. The CPA also hands out ID cards that states one's...	The CPA is meant to prevent crime and not punish crime. It stands for Crime Prevention Association. The CPA organization has made crime nearly impossible through various methods of surveillance and intelligence gathering. The crime was not punished by the CPA but addressed by sending the person to a hospital for expensive treatment to correct and remove the deviance from the person's...

Table 5.1: An example question and four human-written references from SQuALITY. The full references are available in Table A.6 in the appendix.

5.2 Related Work

Story Summarization A common focus of summarization research is on stories and narratives. BookSum [Kryściński et al. 2021] consists of public domain books and summaries of those books, chapters, and paragraphs. Similarly, Ladhak et al. [2020] propose a dataset for summarizing chapters of public domain books. Both of these datasets use summaries scraped from popular study guide websites such as SparkNotes, apparently without an overt license, and thus the datasets cannot be legally distributed. SummScreen Chen et al. [2022] consists of fan-written transcripts of TV episodes paired with Wikipedia and fan-written summaries of those episodes.

Question-Focused Summarization In question-focused summarization (QFS) the summary focuses on a specific aspect of the source text as a way answering a specific question. QFS has received increasing attention from the summarization literature in recent years, and we expect it to be a viable proxy benchmark task for narrative-text summarization broadly. The Debatepedia dataset [Nema et al. 2017] is a found dataset of questions and summary-answers based on articles about social and philosophical issues. FacetSum [Meng et al. 2021] is a found dataset consisting of scientific papers paired with author-written summaries focusing on different aspects of the paper. WikiAsp [Hayashi et al. 2021] and AQuaMuSe [Kulkarni et al. 2020] are two heuristically created, multi-document QFS datasets derived from Wikipedia.

Most similar to our dataset is QMSum [Zhong et al. 2021], a long-document QFS dataset built around meeting transcripts. Like our work, QMSum questions and summaries are composed by writers who have read full transcripts and are guided by a list of question templates. Unlike our work, their primary mechanism for quality control is researcher review of the collected responses, whereas we use a crowdsourcing protocol wherein writers review each other's work.

Long-Form QA QFS is a special case of long-form question answering (LFQA). In LFQA, the inputs are also a question and an input document, and the task is to produce an answer at least one long sentence in length. LFQA answers can draw from a single portion of the document, whereas summaries for QFDS should cover multiple parts of the input document, if not the whole document.

5.3 Dataset Construction

Source Documents Our considerations in selecting a corpus of documents for which to collect summaries are: (1) The documents are long, as document-level tasks are more challenging than paragraph-level ones; (2) The documents can support several substantive summaries, as we will collect multiple summaries per document for cost-efficiency; (3) The documents have a permissive license so they can be easily distributed; (4) The documents are lay-accessible, such that the average college-educated English-fluent speaker can both understand them and confidently evaluate the correctness of summaries derived from them.

We use short stories from Project Gutenberg as they meet all of these desiderata.[4] Specifically, we use a collection of science fiction short stories written in the 1930s–1970s and are between 3000 to 6000 words long. Many of the stories used are also included in the QuALITY [Pang et al. 2021b] dataset, and we coordinate with the QuALITY creators such that stories that appear in both datasets are assigned to the same split. We use the same preprocessing for the stories as used in QuALITY.

Writing For writers to create accurate and high-quality summaries, they need to read the entire story, which takes 20–40 minutes. Rather than asking writers to create one summary per story read, we instead collect multiple summaries per story to amortize the cost of reading across summaries. We solicit multiple summaries by having writers ask questions about different aspects of the story, leading us to create a QFS dataset.

We start each crowdsourcing round by asking writers to read the story and then create questions satisfying two criteria: (1) Questions should require the whole or multiple parts of the story to answer, as opposed to a single sentence or paragraph; (2) To minimize disagreements in evaluation, writers should avoid questions that require speculating substantially beyond the literal text of the story when interpreting themes or symbolism. To assist writers in creating questions satisfying these properties, we provide a list of question templates we expect will meet these properties in most cases, shown in Appendix A.4.1.1. Writers can also write story-specific questions not based on any of these templates so long as they follow the criteria.

For each story, we assign one worker to create four questions. The questions are then answered by four writers, including the original question writer. Each writer also creates a general story summary, framed as answering the question “What is the plot of the story?”, for a total of

[4] https://www.gutenberg.org/

Dataset	Domain	# Examples	Doc. Len	Summ. Len	Multi-ref?	Public?
CNN/DM	news	311k	804	60	✗	✗
XSum	news	226k	438	24	✗	✗
BookSum	fiction, Sparknotes	12k	5102	505	✗	✗
QMSum	meeting transcripts	1808	9067	70	✗	✓
SQuALITY	sci-fi stories	625	5200	237	✓	✓

Table 5.2: Summary statistics for various summarization datasets. For BookSum, we consider the chapter-level version. The number of examples is across all splits. For question-based summarization datasets (SQuALITY and QMSum) we count examples as number of unique document-question pairs. Statistics for datasets are borrowed from original dataset papers; statistics for CNN/DM and XSum were borrowed from Kryściński et al. [2021]. CNN/DM and XSum are often available online in practice, but distributing the dataset is legally questionable.

five questions per story. Responses are required to be 75–500 words long, to avoid copying the text of the story verbatim, and to draw on different parts of the story as much as possible. Writers report that this step takes 40–120 minutes, including time reading the story.

Data Validation After a writing step, for each story, we have five questions with four reference summaries per question. In the second step of each crowdsourcing round, we ask workers to review the summaries to ensure they are high-quality.

As with writing, asking crowdworkers to review the responses is expensive because verifying whether a response is faithful to the story requires having read the entire story. We minimize costs by asking each writer to review the responses of the other three writers. Because the writer has already read the story, they do not need to fully re-read the story, and because they have answered the questions previously, they already have a sense of what constitutes a good response to each question.

In each validation task, we show the reviewer the original story, the set of five questions, and three responses for each question written by other writers. Reviewers first annotate spans of the responses that contain typos or factual errors. Next, they rank the three responses from best to worst. We instruct the reviewers to rank the responses by (1) how well the response correctly answers the question; (2) how well the summary includes all relevant details; (3) how well the response draws from multiple parts of the story, using their judgment to balance the three factors. Writers are informed during the writing step that their responses will be evaluated along these dimensions. Finally, reviewers provide written feedback for each response about how that response could be improved. The feedback is provided to writers between batches of work to help them improve their responses. Reviewers report that this step typically takes 20–30 minutes.

Afterwards, for each question, we compile the individual reviewer rankings into an aggregate ranking. We incentivize high-quality writing by awarding bonus payments to writers based on their response's placement in the overall ranking. We pay $2.50, $1.25, $0.75, $0.50 for ranking first, second, third, and fourth respectively.[5] The average bonus is $1.25 per response, so writers earn an average additional bonus of $6.25 per story. Workers are informed of the bonus structure before writing.

Similarly, we incentivize high-quality reviewing by awarding bonus payments to reviewers based on how well their rankings agree with the aggregate ranking. For each pair of responses, we pay a reviewer a bonus of $0.50 if their ranking of the pair agrees with the aggregate ranking (i.e., if both the aggregate and reviewer's ranking say response A > response B), so reviewers can earn up to $1.50 per question and $7.50 per story. On average, individual reviewers agree with the aggregate ranking on pairwise comparisons 76% of the time, corresponding to an average bonus of $5.57 per story.

Writer Details Because our tasks are very time-consuming and detail-oriented, we eschew crowdsourcing platforms like Amazon Mechanical Turk where eliciting high-quality responses for these types of tasks can be challenging. Instead, we use a small group of skilled writers for long-term contracts, drawing both from Upwork[6] freelancers and undergraduate students from our institution. We hire 11 Upwork writers and 7 undergraduates.[7] Details about the hiring

[5] For ties, we sum the bonuses for the tied positions and distribute them evenly.

[6] https://www.upwork.com/

[7] We use two worker populations due to spending limits on Upwork. The two populations are not mixed: The groups do not review each other's responses.

process and writer populations are in Appendix A.4.1.

5.4 SQuALITY

We present examples from SQuALITY in Table 5.1 and summary statistics of SQuALITY and other summarization datasets in Table 5.2.

Data Size and Splits SQuALITY consists of 100 stories that are split 39/25/36 across the train/-validation/test splits (or, equivalently, 195/125/180 document-question pairs). We assign stories to splits to be consistent with the QuALITY dataset [Pang et al. 2021b], so stories that appear in both datasets are assigned to the same split.

SQuALITY contains a similar number of summaries to QMSum [Zhong et al. 2021], another crowdsourced summarization dataset, but SQuALITY contains four references per example and thus fewer input documents. This difference in allocation arises from the crowdsourcing protocol: In creating SQuALITY, we have writers review each other's work while in creating QMSum, the authors manually review all responses. Protocols wherein workers review each other work are more scalable. Having multiple references per input is useful for model evaluation, as automatic metrics such as ROUGE were originally developed on multi-reference datasets. While naive multi-reference ROUGE still correlates poorly with human judgments of quality for SQuALITY (see Section 5.6), having a diverse set of references opens up opportunities for the development of new evaluation metrics that take into account the diversity of acceptable summaries for a given input, even in the question-focused setting.

Length Documents are an average of 5200 tokens long (std. 522) without punctuation, [8] with a range from 3473 to 6165—similar to the *chapters* version of BookSum, and shorter than QMSum.

Responses average 237 tokens long (std. 133), corresponding to a compression ratio of 95.4%.

[8] We use the `en_core_web_sm` spaCy tokenizer.

Text	N-gram Size			
	1	2	3	4
Diff. story	19.7	2.7	0.1	0.0
Same story	27.4	5.8	1.2	0.4
Same question	33.4	8.7	2.3	0.8
Story	69.4	22.0	5.0	1.7

Table 5.3: (Top) Average percentage of unique n-grams shared between pairs of responses from different sources: two different stories, different questions but the same story, and the same question. (Bottom) Average percentage of unique summary n-grams that also appear in the corresponding story.

The plot summaries have an average length of 442 tokens and are comparable in length to those of BookSum. The other responses are shorter with an average length of 186 tokens, but are still longer than the summaries in QMSum.

Response Diversity We measure summary abstractiveness by computing the percentage of summary n-grams that also appear in the story, shown in Table 5.3. The high recall of 1-grams is unsurprising given the length of the stories, but the low recall of 3- and 4-grams shows that the summaries are highly abstractive.

We next consider the diversity between pairs of responses to the same question. If responses are similar, then collecting multiple references is potentially wasteful. We show in Table 5.3 the average percentage of unique n-grams shared between responses to the same question. The overlap is low: Only 33% of unigrams and less than 10% of bigrams are shared between responses to the same question. This overlap is only slightly higher than the average overlap between responses to completely different stories. The low response overlap highlights the diversity of the summarization task, a property made evident in SQuALITY but not in single-reference datasets.

Model	R-1	R-2	R-L	M	BScore
LED	27.7	5.9	17.7	16.5	82.7
PEGASUS	38.2	9.0	20.2	23.4	84.9
BART	40.2	10.4	20.8	24.5	85.3
BART+DPR	**41.5**	**11.4**	**21.0**	**26.1**	**85.5**
Human*	46.6	12.5	22.7	30.6	86.2

Table 5.4: Automatic evaluation results with ROUGE (1/2/L), METEOR, and BERTScore. LED performs worst and tends to repeat a single sentence. PEGASUS performs substantially better, but lags slightly behind BART. The best performing model is BART+DPR. *The human reference is evaluated with three references while model-generated summaries are evaluated with four references, artificially raising their scores.

5.5 Baselines

5.5.1 Models

We evaluate supervised sequence-to-sequence models on SQuALITY using different pretrained language models as the base model. We implement our baselines using HuggingFace Transformers [Wolf et al. 2020]. We do not explore prompting approaches for summarization with closed-access models. Previous work has found that models can be prompted zero-shot to produce high-quality summaries [Radford et al. 2019; Wu et al. 2021], though public models like GPT-3 do not have the capacity to process full stories from our dataset.

BART BART [Lewis et al. 2020] is a Transformer-based [Vaswani et al. 2017] encoder-decoder model pretrained on a token in-filling objective and a sentence permutation objective. We use `BART-large`, which has a maximum input sequence length of 1024 tokens, so we truncate stories dramatically to fit this simple baseline.

BART+DPR We experiment with an extract-then-summarize baseline. Instead of truncating stories when using BART, we first retrieve story sentences that are most relevant to the question and concatenate them to form the input. We use the pretrained Dense Passage Retriever

[Karpukhin et al. 2020] that encodes the question into a vector representation and retrieves the story sentences that are most similar to the question.

PEGASUS PEGASUS [Zhang et al. 2020] is a Transformer-based encoder-decoder model that is pretrained using an objective designed for summarization. The objective is to predict masked out sentences that are selected to be heuristic pseudo-summaries of the document. PEGASUS is pretrained on sequences of at most length 512, but we follow previous work in finetuning `PEGASUS-large` with a max sequence length of 2048 tokens, truncating stories to fit.

LED Longformer Encoder-Decoder [Beltagy et al. 2020] is an encoder-decoder model where the encoder is a Longformer and the decoder is a Transformer. A Longformer modifies the Transformer architecture with a more efficient self-attention pattern that allows the model to tractably scale to long documents. The LED maximum input length can fit entire stories. We use a context length of 8192 for memory efficiency. The parameters of LED are initialized using BART weights, copied eight times over. We use `LED-base`.

5.5.2 Training

We format example inputs by concatenating the question to the beginning *and* end of the document, separated by a special `[SEP]` token, based on previous work on question-focused summarization [Vig et al. 2021]. Each (story, question, reference) tuple is mapped to a separate training instance, so each (story, question) input is associated with four training examples, one per reference. We finetune models using the AdamW optimizer [Loshchilov and Hutter 2018]. Additional training details are available in Appendix A.4.3.

Model	Corr.	Coverage	Overall
BART	34.8	15.6	18.1
BART+DPR	45.4	24.3	27.9
Human	**94.1**	**88.8**	**91.3**

Table 5.5: Human evaluation results for two models and a human-written response. Corr. stands for correctness. Ratings for each property are averaged across 3 workers, then averaged across questions.

5.5.3 Evaluation

At test time, we generate summaries using beam search with beam width 4. We evaluate the summaries with ROUGE [Lin 2004] and METEOR [Banerjee and Lavie 2005], standard automatic metrics for summarization. We also evaluate with a RoBERTa-large based version of BERTScore [Zhang et al. 2019a], which uses RoBERTa to compute the similarity between references and model generations. For all metrics, we report F1 and handle multiple references by evaluating a candidate against each reference individually, and then taking the max score across references.

5.5.4 Automatic Evaluation Results

We present results using various automatic evaluation metrics in Table 5.4. We observe that LED fails to learn the task and generally produces outputs containing long, repeated sentences. The pathological behavior is reflected in the low ROUGE-1 and ROUGE-2 scores for the model. We hypothesized that the poor performance is because the small dataset size is not enough to finetune the additional positional embeddings. We explored transfer learning approaches where the model was first finetuned on a larger long-context summarization dataset, such as arXiv [Cohan et al. 2018] or GovReport [Huang et al. 2021], and then finetuned on SQuALITY. However, training on intermediate datasets did not fix the issue of degenerate outputs, indicating that the additional positional embeddings were not the bottleneck in the model's performance. Overall, we found that public pretrained models for medium to long input tasks were not effective off the shelf.

PEGASUS, BART, and BART+DPR do substantially better on the task and produce sensible outputs, despite having partial inputs. PEGASUS slightly underperforms the BART variants according to all metrics. BART+DPR outperforms BART with truncated input across all metrics.

Additionally, we evaluate the human references using the automatic metrics by holding one reference out and comparing it with the various metric against the remaining three references. We repeat this process for all references and average the metric score across held-out references. While this use of three references rather than four disadvantages the human references (see Section 5.6), we still find that they score higher than machine outputs.

5.6 Human Evaluation

Automatic metrics for evaluating text summarization have been well-documented as correlating poorly with human judgments of various quality [Schluter 2017; Kryscinski et al. 2019; Durmus et al. 2020]. As such, we accompany automatic evaluation of the baseline systems with human evaluation. We ask workers to rate the quality of outputs from BART and BART+DPR on the test data.

For each task, we show the worker a story and for each of its five questions, two model-generated summaries and a human reference. Workers rate each summary for three properties: correctness, coverage, and overall quality. Each property is rated on a scale from 1-100, similar to direct assessment ratings in machine translation [Bojar et al. 2016]. Workers are instructed to assign ratings that align with their preference rankings between systems [Sakaguchi and Van Durme 2018]. We annotate 20 stories (100 questions) with three Upwork workers per story. Finally, we average property ratings across annotators. The worker details and property definitions are available in Appendix A.4.5.

We present results of the human evaluation in Table 5.5 and sample model generations in Appendix A.4.4. The standard deviations of property ratings across questions are shown in Appendix A.4.5. For all questions and all properties, all human annotators rank the human-written

Metric	Model Only	Human Only	All
ROUGE-1	-7.4	6.8	63.1*
ROUGE-2	-7.8	5.2	42.8*
ROUGE-L	-3.2	14.0	47.8*
METEOR	-11.1	-4.3	54.1*
BERTScore	5.5	0.8	68.7*

Table 5.6: Pearson correlations (multiplied by 100) between automatic evaluation metrics and human judgments of overall quality for three subsets of the human evaluation data: only model-generated summaries ('model only'), only human written summaries ('human only'), and both ('all'). Correlations are only significant (*) when considering all summaries.

response as better than the model responses. The human-written response has an average rating around or above 90 for all three properties. On the other hand, BART and BART+DPR have an average rating below 50 for all three properties, substantially below corresponding ratings for the human response. Across all three properties, BART+DPR is ranked as better than BART on 70% of examples. The models receive the highest rating on the correctness property among all properties. Upon inspecting the model generations, we partly attribute these relatively high ratings to the fact that the model-generated responses are fairly generic and devoid of specific details. This lack of specificity is reflected in the especially low coverage ratings of the model-generated summaries. Overall, we conclude that fully-public automatic summarization systems still lag significantly behind human writers.

Correlation Between Automatic and Human Evaluations We next consider the correlations between automatic and human evaluations for three subsets of the collected data: only model-written summaries (200 summaries), only human-written summaries (100 summaries), and all summaries. We present the correlations with the judgments of overall quality for these subsets in Table 5.6.

When considering all summaries, all metrics have a substantial positive correlation with the human judgments of overall quality. However, these appear to mostly reflect the fact that the automatic metrics rank human-written summaries as better than model-written ones: When considering only model-written summaries or only human-written summaries, the correlations are dramatically weaker and are in no cases significant.

The weak correlations in these settings point to the brittleness of using these automatic metrics when comparing the outputs of two automatic summarization systems, where metric values will similarly be in a narrow range. In light of these findings, we caution against relying on automatic metrics to measure system quality on SQuALITY and instead rely on human evaluation of model outputs.

Multi-Reference Automatic Metrics We next consider whether having multiple references improves the correlation of automatic evaluation metrics. ROUGE was originally developed on multi-reference datasets, but recent summarization datasets are predominantly single reference. This mismatch may contribute to the poor correlation of ROUGE with human judgments of quality for these datasets [Pang et al. 2021a; Pagnoni et al. 2021; Scialom et al. 2021, i.a.]. We use the multiple references of SQuALITY to measure the effect of varying the number of references used in automatic metrics on the correlation with human judgments.

We find that using fewer references when computing the automatic evaluation metrics does not substantially change the correlations with human judgments. To demonstrate why, we show the average and maximum metric values for each automatic metric in Table 5.7. We observe that for all metrics considered, the maximum value of the metric is relatively close to the average metric value across references. Despite having diverse references, the metric values are similar across references. Thus, using multiple references does not improve correlations between automatic metrics and human judgments of overall quality. However, we note that simply taking the maximum metric value over references is relatively simple, and that there may be more sophisticated ways to use the diverse references to compute generation quality.

Metric	Avg.	Max.	Δ
ROUGE-1	37.9	41.5	3.6
ROUGE-2	8.7	11.4	2.1
ROUGE-L	18.8	21.0	2.2
METEOR	22.7	26.1	3.4
BERTScore	84.8	85.5	0.7

Table 5.7: Average and maximum metric value across the four references for BART+DPR when considering only a single reference at a time.

5.7 Conclusion

We present SQuALITY, a long-input dataset for abstractive question-focused summarization. Because the summaries are crowdsourced rather than found, we can use input documents that are of an accessible domain and under an open license to avoid common issues with existing summarization datasets. Our crowdsourcing protocol gives multiple summaries and references per input while making the cost of data collection more tractable.

Baseline results with competitive public medium-scale pretrained models suggest that the dataset remains beyond the capabilities of such systems. Our best performing model is an extract-then-summarize model where we use the questions to retrieve story sentences as input. The performance of proprietary larger-scale models remains an open question, and may depend significantly on whether such models can process the full stories without truncation.

Given the poor correlation of existing automatic metrics with human judgments of model outputs, we expect that automatic metrics will provide a very weak signal for progress on SQuALITY. We recommend that researchers using SQuALITY evaluate their summarization systems by having human annotators read a selection of our source stories and compare model outputs on those stories. To facilitate this, we will make our templates for human evaluation available, though creating efficient and effective methods for evaluating summaries of long input documents remains an open issue.

6 | Conclusion

In this dissertation, we explored the effects that pretrained language models have had on how we evaluate NLP systems. In Chapter 2, we introduced a standardized benchmark for evaluating the generalization ability of NLP systems by their ability to perform many diverse downstream NLP tasks. In Chapter 3, we updated this benchmark with a more challenging set of downstream tasks in light of rapid progress made by pretrained language models, and discussed steps we took and future directions for developing more challenging general-purpose language understanding benchmarks. In Chapter 4, we turn pretrained language models on themselves to develop an improved evaluation metric for measuring faithfulness of automatically generated summaries that leverages the improved question answering abilities of pretrained language models. Finally, in Chapter 5, we developed high-quality evaluation data for automatic text summarization by crowdsourcing summaries rather than relying on found data on the web.

Our work points to a number of open problems remaining in evaluating NLP models. As model capabilities continue to improve, they become better at exploit biases and data artifacts in existing datasets, necessitating the creation of higher quality data and more challenging tasks. One promising approach for developing such data is incorporating NLP models in the data creation process, such as to discover noisy examples in datasets [Swayamdipta et al. 2020] or by assisting crowdworkers with machine learning models during the crowdsourcing process [Liu et al. 2022; Bartolo et al. 2021; Saunders et al. 2022].

For text generation problems specifically, there has been increasing scrutiny paid towards human evaluation of model outputs [Howcroft et al. 2020; Iskender et al. 2021; Smith et al. 2022, i.a.]. Human evaluation has traditionally been regarded as the gold standard in evaluating text generation models. However, recent studies have identified inconsistent task definitions and standards between papers. Thus, a critical open problem is the development of consistent and high-quality human evaluation protocols for effective evaluation of NLG systems. Annotations collected from such protocols could be used to train the next generation of automatic model-based evaluation metrics, in the vein of Sellam et al. [2020a] or Stiennon et al. [2020], that are trained to mimic human judgments of complex properties.

Overall, incorporating the growing capabilities of pretrained language models to evaluate the growing capabilities of language models is a promising future direction.

A | Appendix

A.1 Appendices for GLUE

A.1.1 Additional Data Details

A.1.1.1 Dataset Construction

QNLI To construct a balanced dataset, we select all pairs in which the most similar sentence to the question was *not* the answer sentence, as well as an equal amount of cases in which the correct sentence was the most similar to the question, but another distracting sentence was a close second. Our similarity metric is based on CBoW representations with pre-trained GloVe embeddings. This approach to converting pre-existing datasets into NLI format is closely related to recent work by White et al. [2017], as well as to the original motivation for textual entailment presented by Dagan et al. [2006]. Both argue that many NLP tasks can be productively reduced to textual entailment.

A.1.1.2 Diagnostic Data

We show the full label set used to tag the diagnostic set in Table A.1.

Coarse-Grained Categories	Fine-Grained Categories
Lexical Semantics	Lexical Entailment, Morphological Negation, Factivity, Symmetry/Collectivity, Redundancy, Named Entities, Quantifiers
Predicate-Argument Structure	Core Arguments, Prepositional Phrases, Ellipsis/Implicits, Anaphora/Coreference Active/Passive, Nominalization, Genitives/Partitives, Datives, Relative Clauses, Coordination Scope, Intersectivity, Restrictivity
Logic	Negation, Double Negation, Intervals/Numbers, Conjunction, Disjunction, Conditionals, Universal, Existential, Temporal, Upward Monotone, Downward Monotone, Non-Monotone
Knowledge	Common Sense, World Knowledge

Table A.1: The types of linguistic phenomena annotated in the diagnostic dataset, organized under four major categories.

A.1.2 Additional Baseline Details

A.1.2.1 Attention Mechanism

We implement our attention mechanism as follows: Given sequences of hidden states $u_1, u_2, \ldots, u_M$ and $v_1, v_2, \ldots, v_N$, we first compute matrix H where $H_{ij} = u_i \cdot v_j$. For each u_i, we get attention weights α_i by taking a softmax over the i^{th} row of H, and get the corresponding context vector $\tilde{v}_i = \sum_j \alpha_{ij} v_j$ by taking the attention-weighted sum of the v_j. We pass a second BiLSTM with max pooling over the sequence $[u_1; \tilde{v}_1], \ldots [u_M; \tilde{v}_M]$ to produce u'. We process the v_j vectors analogously to obtain v'. Finally, we feed $[u'; v'; |u' - v'|; u' * v']$ into a classifier.

A.1.2.2 Training

We train our models with the BiLSTM sentence encoder and post-attention BiLSTMs shared across tasks, and classifiers trained separately for each task. For each training update, we sample a task to train with a probability proportional to the number of training examples for each task. We scale each task's loss inversely proportional to the number of examples for that task, which we found to improve overall performance. We train our models with Adam [Kingma and Ba 2014] with initial learning rate 10^{-3}, batch size 128, and gradient clipping. We use macro-average score over all tasks as our validation metric, and perform a validation check every 10k updates. We divide the learning rate by 5 whenever validation performance does not improve. We stop training when the learning rate drops below 10^{-5} or performance does not improve after 5 validation checks.

A.1.2.3 Sentence Representation Models

We evaluate the following sentence representation models:

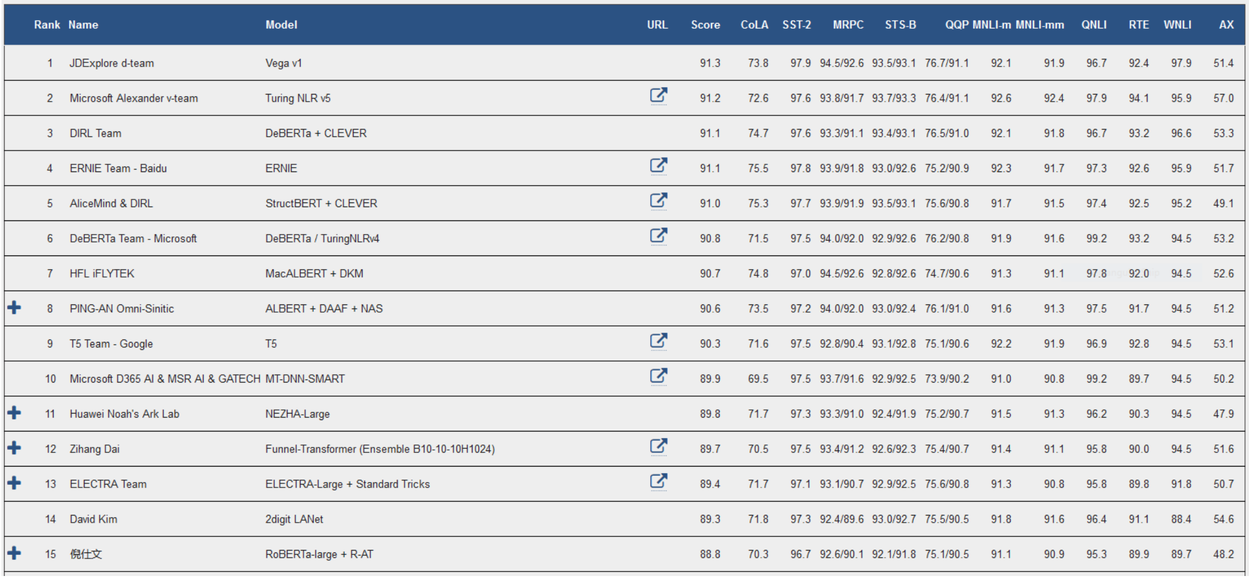
1. CBoW, the average of the GloVe embeddings of the tokens in the sentence.

2. Skip-Thought [Kiros et al. 2015], a sequence-to-sequence(s) model trained to generate the previous and next sentences given the middle sentence. We use the original pre-trained model[1] trained on sequences of sentences from the Toronto Book Corpus (Zhu et al. 2015, TBC).

3. InferSent [Conneau et al. 2017], a BiLSTM with max-pooling trained on MNLI and SNLI.

4. DisSent [Nie et al. 2017], a BiLSTM with max-pooling trained to predict the discourse marker (*because*, *so*, etc.) relating two sentences on data derived from TBC. We use the variant trained for eight-way classification.

5. GenSen [Subramanian et al. 2018], a sequence-to-sequence model trained on a variety of supervised and unsupervised objectives. We use the variant of the model trained on both MNLI and SNLI, the Skip-Thought objective on TBC, and a constituency parsing objective on the Billion Word Benchmark.



	Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
	1	JDExplore d-team	Vega v1		91.3	73.8	97.9	94.5/92.6	93.5/93.1	76.7/91.1	92.1	91.9	96.7	92.4	97.9	51.4
	2	Microsoft Alexander v-team	Turing NLR v5	↗	91.2	72.6	97.6	93.8/91.7	93.7/93.3	76.4/91.1	92.6	92.4	97.9	94.1	95.9	57.0
	3	DIRL Team	DeBERTa + CLEVER		91.1	74.7	97.6	93.3/91.1	93.4/93.1	76.5/91.0	92.1	91.8	96.7	93.2	96.6	53.3
	4	ERNIE Team - Baidu	ERNIE	↗	91.1	75.5	97.8	93.9/91.8	93.0/92.6	75.2/90.9	92.3	91.7	97.3	92.6	95.9	51.7
	5	AliceMind & DIRL	StructBERT + CLEVER	↗	91.0	75.3	97.7	93.9/91.9	93.5/93.1	75.6/90.8	91.7	91.5	97.4	92.5	95.2	49.1
	6	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4	↗	90.8	71.5	97.5	94.0/92.0	92.9/92.6	76.2/90.8	91.9	91.6	99.2	93.2	94.5	53.2
	7	HFL iFLYTEK	MacALBERT + DKM		90.7	74.8	97.0	94.5/92.6	92.8/92.6	74.7/90.6	91.3	91.1	97.8	92.0	94.5	52.6
+	8	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6	73.5	97.2	94.0/92.0	93.0/92.4	76.1/91.0	91.6	91.3	97.5	91.7	94.5	51.2
	9	T5 Team - Google	T5	↗	90.3	71.6	97.5	92.8/90.4	93.1/92.8	75.1/90.6	92.2	91.9	96.9	92.8	94.5	53.1
	10	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART	↗	89.9	69.5	97.5	93.7/91.6	92.9/92.5	73.9/90.2	91.0	90.8	99.2	89.7	94.5	50.2
+	11	Huawei Noah's Ark Lab	NEZHA-Large		89.8	71.7	97.3	93.3/91.0	92.4/91.9	75.2/90.7	91.5	91.3	96.2	90.3	94.5	47.9
+	12	Zihang Dai	Funnel-Transformer (Ensemble B10-10-10H1024)	↗	89.7	70.5	97.5	93.4/91.2	92.6/92.3	75.4/90.7	91.4	91.1	95.8	90.0	94.5	51.6
+	13	ELECTRA Team	ELECTRA-Large + Standard Tricks	↗	89.4	71.7	97.1	93.1/90.7	92.9/92.5	75.6/90.8	91.3	90.8	95.8	89.8	91.8	50.7
	14	David Kim	2digit LANet		89.3	71.8	97.3	92.4/89.6	93.0/92.7	75.5/90.5	91.8	91.6	96.4	91.1	88.4	54.6
+	15	倪仕文	RoBERTa-large + R-AT		88.8	70.3	96.7	92.6/90.1	92.1/91.8	75.1/90.5	91.1	90.9	95.3	89.9	89.7	48.2

Figure A.1: The benchmark website leaderboard. An expanded view shows additional details about each submission, including a brief prose description and parameter count.

A.1.3 Benchmark Website Details

GLUE's online platform is built using React, Redux and TypeScript. We use Google Firebase for data storage and Google Cloud Functions to host and run our grading script when a submission is made. Figure A.1 shows the visual presentation of our baselines on the leaderboard.

[1] github.com/ryankiros/skip-thoughts

A.2 Appendices for SuperGLUE

A.2.1 Development Set Results

In Table A.2, we present results of the baselines on the SuperGLUE tasks development sets.

Table A.2: Baseline performance on the SuperGLUE development.

Model **Metrics**	**Avg**	**BoolQ** **Acc.**	**CB** **Acc./F1**	**COPA** **Acc.**	**MultiRC** **F1$_a$/EM**	**ReCoRD** **F1/EM**	**RTE** **Acc.**	**WiC** **Acc.**	**WSC** **Acc.**
Most Frequent Class	47.7	62.2	50.0/22.2	55.0	59.9/ 0.8	32.4/ 31.5	52.7	50.0	63.5
CBOW	47.7	62.4	71.4/49.6	63.0	20.3/ 0.3	14.4/ 13.8	54.2	55.3	61.5
BERT	72.2	77.7	94.6/93.7	69.0	70.5/ 24.7	70.6/ 69.8	75.8	74.9	68.3
BERT++	74.6	80.1	96.4/95.0	78.0	70.5/ 24.7	70.6/ 69.8	82.3	74.9	68.3

A.2.2 Performance on GLUE Diagnostics

Figure A.2 shows the performance on the GLUE diagnostics dataset for systems submitted to the public leaderboard.

A.2.3 Human Performance Estimation

For collecting data to establish human performance on the SuperGLUE tasks, we follow a two step procedure where we first provide some training to the crowd workers before they proceed to annotation. For both steps and all tasks, the average pay rate is $23.75/hr.[2]

In the training phase, workers are provided with instructions on the task, linked to an FAQ page, and are asked to annotate up to 30 examples from the development set. After answering each example, workers are also asked to check their work against the provided ground truth label. After the training phase is complete, we provide the qualification to work on the annotation phase to all workers who annotated a minimum of five examples, i.e. completed five HITs during

[2] This estimate is taken from `https://turkerview.com`.

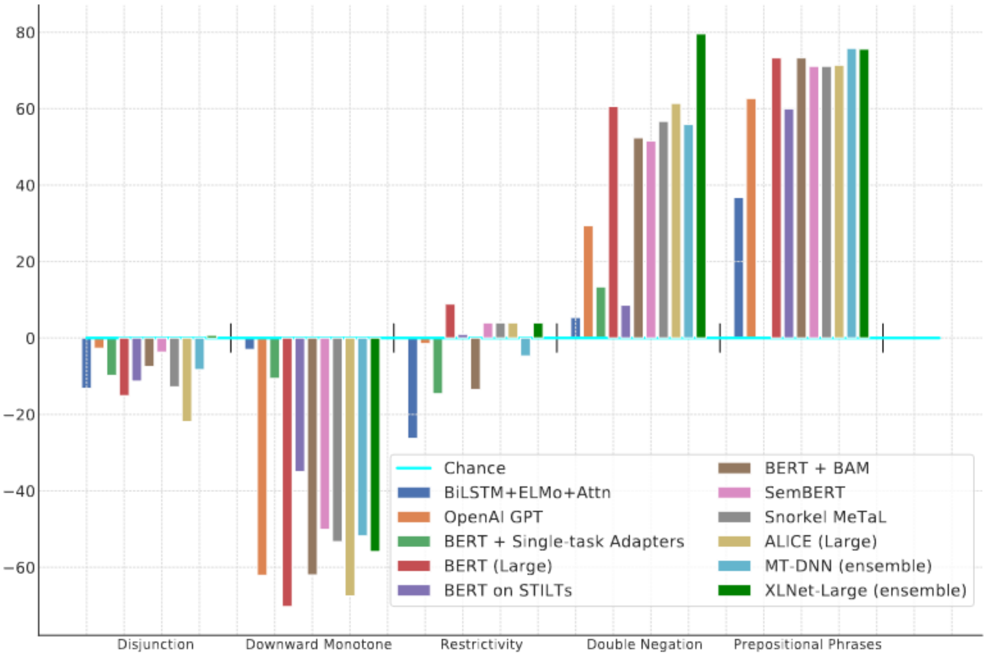


Figure A.2: Performance of GLUE submissions on selected diagnostic categories, reported using the R_3 metric scaled up by 100, as in Wang et al. [2019b, see paper for a description of the categories]. Some initially difficult categories, like double negation, saw gains from advances on GLUE, but others remain hard (restrictivity) or even adversarial (disjunction, downward monotone).

training and achieved performance at, or above the median performance across all workers during training.

In the annotation phase, workers are provided with the same instructions as the training phase, and are linked to the same FAQ page. The instructions for all tasks are provided in Appendix A.2.3. For the annotation phase we randomly sample 100 examples from the task's test set, with the exception of WSC where we annotate the full test set. For each example, we collect annotations from five workers and take a majority vote to estimate human performance. For additional details, see Appendix A.2.3.3.

A.2.3.1 Training Phase Instructions

In the training step, we provide workers with brief instructions about the training phase. An example of these instructions is given Table A.3. These training instructions are the same across

Table A.3: Task-specific instructions for Choice of Plausible Alternatives (COPA). These instructions were provided during both training and annotation phases.

Plausible Answer Instructions

The New York University Center for Data Science is collecting your answers for use in research on computer understanding of English. Thank you for your help!

We will present you with a prompt sentence and a question. The question will either be about what caused the situation described in the prompt, or what a possible effect of that situation is. We will also give you two possible answers to this question. Your job is to decide, given the situation described in the prompt, which of the two options is a more plausible answer to the question:

In the following example, option 1. is a more plausible answer to the question about what caused the situation described in the prompt,

> *The girl received a trophy.*
>
> *What's the CAUSE for this?*
>
> 1. *She won a spelling bee.*
> 2. *She made a new friend.*

In the following example, option 2. is a more plausible answer the question about what happened because of the situation described in the prompt,

> *The police aimed their weapons at the fugitive.*
>
> *What happened as a RESULT?*
>
> 1. *The fugitive fell to the ground.*
> 2. *The fugitive dropped his gun.*

If you have any more questions, please refer to our FAQ page.

---

tasks, only the task name in the instructions is changed.

A.2.3.2 Task Instructions

During training and annotation for each task, we provide workers with brief instructions tailored to the task. We also link workers to an FAQ page for the task. Tables A.3, A.4, and A.5 show the instructions we used for COPA, CommitmentBank, and WSC respectively.

We collected data to produce conservative estimates for human performance on several tasks

Table A.4: Task-specific instructions for Commitment Bank. These instructions were provided during both training and annotation phases.

Speaker Commitment Instructions

The New York University Center for Data Science is collecting your answers for use in research on computer understanding of English. Thank you for your help!

We will present you with a prompt taken from a piece of dialogue, this could be a single sentence, a few sentences, or a short exchange between people. Your job is to figure out, based on this first prompt (on top), how certain the speaker is about the truthfulness of the second prompt (on the bottom). You can choose from a 7 point scale ranging from (1) completely certain that the second prompt is true to (7) completely certain that the second prompt is false. Here are examples for a few of the labels:

Choose 1 (certain that it is true) if the speaker from the first prompt definitely believes or knows that the second prompt is true. For example,

> *"What fun to hear Artemis laugh. She's such a serious child. I didn't know she had a sense of humor."*
>
> *"Artemis had a sense of humor"*

Choose 4 (not certain if it is true or false) if the speaker from the first prompt is uncertain if the second prompt is true or false. For example,

> *"Tess is committed to track. She's always trained with all her heart and soul. One can only hope that she has recovered from the flu and will cross the finish line."*
>
> *"Tess crossed the finish line."*

Choose 7 (certain that it is false) if the speaker from the first prompt definitely believes or knows that the second prompt is false. For example,

> *"Did you hear about Olivia's chemistry test? She studied really hard. But even after putting in all that time and energy, she didn't manage to pass the test".*
>
> *"Olivia passed the test."*

If you have any more questions, please refer to our FAQ page.

that we did not ultimately include in our benchmark, including GAP [Webster et al. 2018], PAWS [Zhang et al. 2019b], Quora Insincere Questions,[3] Ultrafine Entity Typing [Choi et al. 2018b], and Empathetic Reactions datasets [Buechel et al. 2018].

[3] https://www.kaggle.com/c/quora-insincere-questions-classification/data

Table A.5: Task-specific instructions for Winograd Schema Challenge (WSC). These instructions were provided during both training and annotation phases.

Winograd Schema Instructions

The New York University Center for Data Science is collecting your answers for use in research on computer understanding of English. Thank you for your help!

We will present you with a sentence that someone wrote, with one bolded pronoun. We will then ask if you if the pronoun refers to a specific word or phrase in the sentence. Your job is to figure out, based on the sentence, if the bolded pronoun refers to this selected word or phrase:

Choose Yes if the pronoun refers to the selected word or phrase. For example,

> *"I put the cake away in the refrigerator. It has a lot of butter in it."*
>
> *Does **It** in "It has a lot" refer to **cake**?*

Choose No if the pronoun does not refer to the selected word or phrase. For example,

> *"The large ball crashed right through the table because it was made of styrofoam."*
>
> *Does **it** in "it was made" refer to **ball**?*

If you have any more questions, please refer to our FAQ page.

A.2.3.3 Task Specific Details

For WSC and COPA we provide annotators with a two way classification problem. We then use majority vote across annotations to calculate human performance.

CommitmentBank We follow the authors in providing annotators with a 7-way classification problem. We then collapse the annotations into 3 classes by using the same ranges for bucketing used by [de Marneffe et al. 2019]. We then use majority vote to get human performance numbers on the task.

Furthermore, for training on CommitmentBank we randomly sample examples from the low inter-annotator agreement portion of the CommitmentBank data that is not included in the benchmark version of the task. These low agreement examples are generally harder to classify since they are more ambiguous.

Diagnostic Dataset Since the diagnostic dataset does not come with accompanying training data, we train our workers on examples from RTE's development set. RTE is also a textual entailment task and is the most closely related task in the main benchmark. Providing the crowd workers with training on RTE enables them to learn label definitions which should generalize to the diagnostic dataset.

Ultrafine Entity Typing We cast the task into a binary classification problem to make it an easier task for non-expert crowd workers. We work in cooperation with the authors of the dataset [Choi et al. 2018b] to do this reformulation: We give workers one possible tag for a word or phrase and asked them to classify the tag as being applicable or not.

The authors used WordNet [Miller 1995] to expand the set of labels to include synonyms and hypernyms from WordNet. They then asked five annotators to validate these tags. The tags from this validation had high agreement, and were included in the publicly available Ultrafine Entity Typing dataset,[4] This constitutes our set of positive examples. The rest of the tags from the validation procedure that are not in the public dataset constitute our negative examples.

GAP For the Gendered Ambiguous Pronoun Coreference task [GAP, Webster et al. 2018], we simplified the task by providing noun phrase spans as part of the input, thus reducing the original structure prediction task to a classification task. This task was presented to crowd workers as a three way classification problem: Choose span A, B, or neither.

A.2.4 Excluded Tasks

In this section we provide some examples of tasks that we evaluated for inclusion but ultimately could not include. We report on these excluded tasks only with the permission of their authors. We turned down many medical text datasets because they are usually only accessible with explicit

[4] https://homes.cs.washington.edu/~eunsol/open_entity.html

permission and credentials from the data owners.

Tasks like QuAC [Choi et al. 2018a] and STREUSLE [Schneider and Smith 2015] differed substantially from the format of other tasks in our benchmark, which we worried would incentivize users to spend significant effort on task-specific model designs, rather than focusing on general-purpose techniques. It was challenging to train annotators to do well on Quora Insincere Questions [5], Empathetic Reactions [Buechel et al. 2018], and a recast version of Ultra-Fine Entity Typing [Choi et al. 2018b, see Appendix A.2.3.3 for details], leading to low human performance. BERT achieved very high or superhuman performance on Query Well-Formedness [Faruqui and Das 2018], PAWS [Zhang et al. 2019b], Discovering Ongoing Conversations [Zanzotto and Ferrone 2017], and GAP [Webster et al. 2018].

During the process of selecting tasks for our benchmark, we collected human performance baselines and run BERT-based machine baselines for some tasks that we ultimately excluded from our task list. We chose to exclude these tasks because our BERT baseline performs better than our human performance baseline or if the gap between human and machine performance is small.

On Quora Insincere Questions our BERT baseline outperforms our human baseline by a small margin: an F1 score of 67.2 versus 66.7 for BERT and human baselines respectively. Similarly, on the Empathetic Reactions dataset, BERT outperforms our human baseline, where BERT's predictions have a Pearson correlation of 0.45 on empathy and 0.55 on distress, compared to 0.45 and 0.35 for our human baseline. For PAWS-Wiki, we report that BERT achieves an accuracy of 91.9%, while our human baseline achieved 84% accuracy. These three tasks are excluded from the benchmark since our, admittedly conservative, human baselines are worse than machine performance. Our human performance baselines are subject to the clarity of our instructions (all instructions can be found in Appendix A.2.3), and crowd workers engagement and ability.

For the Query Well-Formedness task, the authors set an estimate human performance at 88.4% accuracy. Our BERT baseline model reaches an accuracy of 82.3%. While there is a positive gap

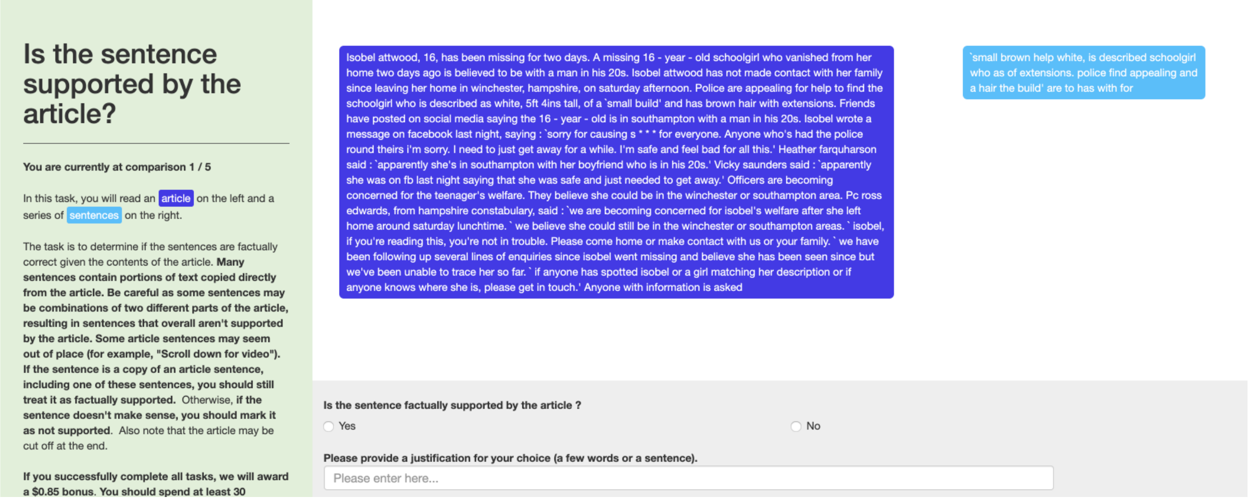
[5] https://www.kaggle.com/c/quora-insincere-questions-classification/data

on this task, the gap was smaller than we were were willing to tolerate. Similarly, on our recast version of the Ultrafine Entity Typing, we observe too small a gap between human (60.2 F1) and machine performance (55.0 F1). Our recasting for this task is described in Appendix A.2.3.2. On GAP, when taken as a classification problem without the related task of span selection (details in A.2.3.2), BERT performs (91.0 F1) comparably to our human baseline (94.9 F1). Given this small margin, we also exclude GAP.

On Discovering Ongoing Conversations, our BERT baseline achieves an F1 of 51.9 on a version of the task cast as sentence pair classification (given two snippets of texts from plays, determine if the second snippet is a continuation of the first). This dataset is very class imbalanced (90% negative), so we also experimented with a class-balanced version on which our BERT baselines achieves 88.4 F1. Qualitatively, we also found the task challenging for humans as there was little context for the text snippets and the examples were drawn from plays using early English. Given this fairly high machine performance and challenging nature for humans, we exclude this task from our benchmark.

A.3 Appendices for QAGS

A.3.1 Human Evaluation Task Design



Is the sentence supported by the article?

You are currently at comparison 1 / 5

In this task, you will read an article on the left and a series of sentences on the right.

The task is to determine if the sentences are factually correct given the contents of the article. **Many sentences contain portions of text copied directly from the article. Be careful as some sentences may be combinations of two different parts of the article, resulting in sentences that overall aren't supported by the article. Some article sentences may seem out of place (for example, "Scroll down for video"). If the sentence is a copy of an article sentence, including one of these sentences, you should still treat it as factually supported.** Otherwise, **if the sentence doesn't make sense, you should mark it as not supported**. Also note that the article may be cut off at the end.

If you successfully complete all tasks, we will award a $0.85 bonus. You should spend at least 30

Isobel attwood, 16, has been missing for two days. A missing 16 - year - old schoolgirl who vanished from her home two days ago is believed to be with a man in his 20s. Isobel attwood has not made contact with her family since leaving her home in winchester, hampshire, on saturday afternoon. Police are appealing for help to find the schoolgirl who is described as white, 5ft 4ins tall, of a `small build' and has brown hair with extensions. Friends have posted on social media saying the 16 - year - old is in southampton with a man in his 20s. Isobel wrote a message on facebook last night, saying : `sorry for causing s * * * for everyone. Anyone who's had the police round theirs i'm sorry. I need to just get away for a while. I'm safe and feel bad for all this.' Heather farquharson said : `apparently she's in southampton with her boyfriend who is in his 20s.' Vicky saunders said : `apparently she was on fb last night saying that she was safe and just needed to get away.' Officers are becoming concerned for the teenager's welfare. They believe she could be in the winchester or southampton area. Pc ross edwards, from hampshire constabulary, said : `we are becoming concerned for isobel's welfare after she left home around saturday lunchtime. ` we believe she could still be in the winchester or southampton areas. ` isobel, if you're reading this, you're not in trouble. Please come home or make contact with us or your family. ` we have been following up several lines of enquiries since isobel went missing and believe she has been seen since but we've been unable to trace her so far. ` if anyone has spotted isobel or a girl matching her description or if anyone knows where she is, please get in touch.' Anyone with information is asked

`small brown help white, is described schoolgirl who as of extensions. police find appealing and a hair the build' are to has with for

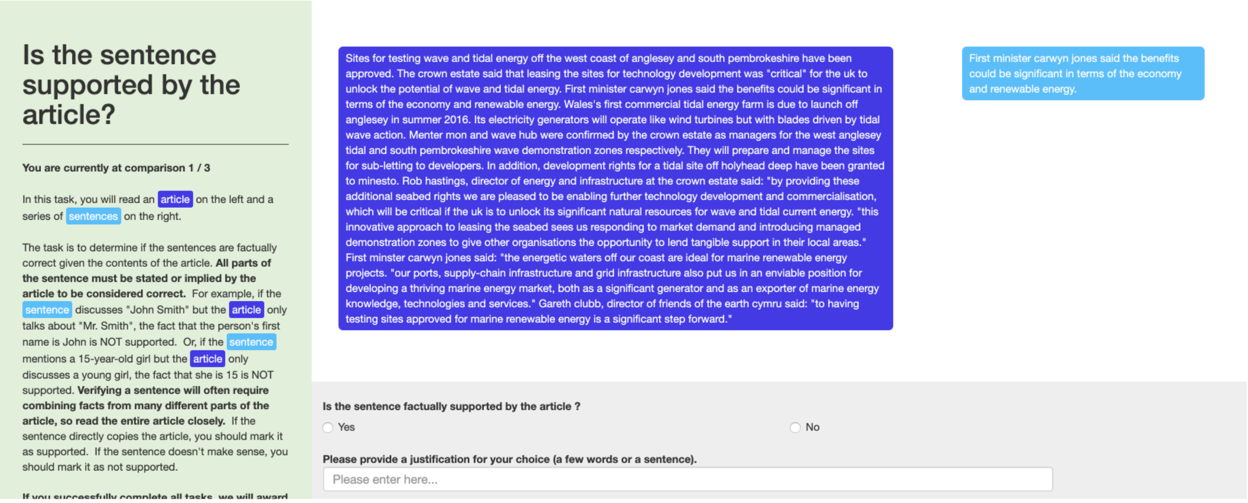
Is the sentence factually supported by the article ?

○ Yes ○ No

Please provide a justification for your choice (a few words or a sentence).

Please enter here...

Figure A.3: Annotation interface and instructions for CNN/DM factual consistency task.



Is the sentence supported by the article?

You are currently at comparison 1 / 3

In this task, you will read an article on the left and a series of sentences on the right.

The task is to determine if the sentences are factually correct given the contents of the article. **All parts of the sentence must be stated or implied by the article to be considered correct.** For example, if the sentence discusses "John Smith" but the article only talks about "Mr. Smith", the fact that the person's first name is John is NOT supported. Or, if the sentence mentions a 15-year-old girl but the article only discusses a young girl, the fact that she is 15 is NOT supported. **Verifying a sentence will often require combining facts from many different parts of the article, so read the entire article closely.** If the sentence directly copies the article, you should mark it as supported. If the sentence doesn't make sense, you should mark it as not supported.

If you successfully complete all tasks, we will award

Sites for testing wave and tidal energy off the west coast of anglesey and south pembrokeshire have been approved. The crown estate said that leasing the sites for technology development was "critical" for the uk to unlock the potential of wave and tidal energy. First minister carwyn jones said the benefits could be significant in terms of the economy and renewable energy. Wales's first commercial tidal energy farm is due to launch off anglesey in summer 2016. Its electricity generators will operate like wind turbines but with blades driven by tidal wave action. Menter mon and wave hub were confirmed by the crown estate as managers for the west anglesey tidal and south pembrokeshire wave demonstration zones respectively. They will prepare and manage the sites for sub-letting to developers. In addition, development rights for a tidal site off holyhead deep have been granted to minesto. Rob hastings, director of energy and infrastructure at the crown estate said: "by providing these additional seabed rights we are pleased to be enabling further technology development and commercialisation, which will be critical if the uk is to unlock its significant natural resources for wave and tidal current energy. "this innovative approach to leasing the seabed sees us responding to market demand and introducing managed demonstration zones to give other organisations the opportunity to lend tangible support in their local areas." First minster carwyn jones said: "the energetic waters off our coast are ideal for marine renewable energy projects. "our ports, supply-chain infrastructure and grid infrastructure also put us in an enviable position for developing a thriving marine energy market, both as a significant generator and as an exporter of marine energy knowledge, technologies and services." Gareth clubb, director of friends of the earth cymru said: "to having testing sites approved for marine renewable energy is a significant step forward."

First minister carwyn jones said the benefits could be significant in terms of the economy and renewable energy.

Is the sentence factually supported by the article ?

○ Yes ○ No

Please provide a justification for your choice (a few words or a sentence).

Please enter here...

Figure A.4: Annotation interface and instructions for XSUM factual consistency task.

We restrict our pool of workers to US-based workers. Workers are required to have at least 1000 approved HITs with an acceptance rate of at least 98%.

The base reward for our task is $0.15. For each summary, we include automatic quality checks including

- Time checks: workers who complete the task under 30s fail the check
- Attention checks: we include exact copies of article sentences and corrupted mixtures of two article sentences as positive and negative control task. If a worker fails to answer both of these examples correctly, they fail the check
- Explanation checks: For each sentence in the summary, the worker is required to provide a short explanation of their decision

If a worker passes all checks, they are awarded a $0.85 bonus, totalling $1.00 per correct annotation. According to `turkerview.com`, workers of our HIT are paid well in excess of $15.00 on average.

We show our annotation interfaces for the annotation task for CNN/DM and XSUM respectively in Figures A.3 and A.4. We use slightly different instructions to accommodate for the quirks of each dataset. For XSUM, we prepend the reference "summary" back onto the source article, as without it, workers were struggling to identify factual inconsistencies.

A.3.2 Model and Generation Details

Question Generation We fine-tune BART for question generation using the same tuning hyperparameters as the original work. We optimize label smoothed cross entropy with smoothing parameter 0.1 [Pereyra et al. 2017] and a peak learning rate of 2e-5. We optimize for 100k steps with 5k warmup steps, and use the model with the best perplexity on the development set.

To turn NewsQA into an answer conditional QG dataset, we concatenate the answer to the source article with a special marker token in between. We then concatenate another special marker token and the question. At test time, we get 10 named entities and noun phrases as answer candidates using the `en-web-sm` spaCy model. We randomly sample 10 if there are more than 10, and randomly duplicate some answers if there are fewer than 10. The model predicts the question after seeing an answer and the article.

During decoding, we use beam search with beam size 10, length penalty 1.0, and trigram repetition blocking. Generations have minimum length 8 and max length 60.

To filter the questions, we first use simple heuristics, including removing

- everything after the first question mark in a question
- exact duplicates
- questions shorter than three tokens long

For the remaining questions, we use our QA model to answer each question and we remove questions for which the QA model deems unanswerable. We then take the top 20 most probable questions, random sampling some of the filtered questions if there were too few.

Question Answering We fine-tune BERT for question answering following the original work. Similar to the QG setting, we append the question and answer to the source article with intervening special marker tokens. We optimize using AdamW [Loshchilov and Hutter 2018] with initial learning rate 5e-5. We train for 3 epochs, with a warmup ratio of 0.1. We use the model with the best development set performance.

A.4 Appendices for SQuALITY

A.4.1 Crowdsourcing Details

We hire Upworker workers and local undergraduates as writers for our data collection pipeline. Most writers create 20–40 responses for the dataset, although five authors submitted 10 or fewer responses. All writers are informed that their writing will be released publicly for use in AI development.

Our Upwork writers are typically US-based native English speakers. Many of them are college-educated, frequently with degrees in the humanities and prior experience in professional copywriting and editing. We found workers for our task by posting an open call on Upwork to participate in a paid interview. In the interview, applicants review an example writing task with sample questions and responses, and then complete a practice writing task. We hired the top 33% of writers based on their performance on the interview task after manually reviewing their responses. We pay Upwork workers $13 and $8 for each writing and reviewing task respectively, with additional opportunities for bonuses described above.

The undergraduates we hire are all English-fluent and come from diverse nationalities and areas of study—the smaller and more junior pool of applicants prevents us from focusing as much on relevant experience as we do with Upwork. Students are paid a constant $20/hr.[6] Students are hired based on relevant experience and writing samples. After they are hired, we show them the same example task and have them do the practice writing task that we showed the Upwork workers.

A.4.1.1 Question Templates

We provide the following question templates to the writers:

[6] Due to the structure of student employment contracts, we are unable to pay students using the bonus payment structure and we instead periodically manually review their responses to ensure they are high-quality.

- What is the plot of the story?
- What happens to [character X] throughout the story?
- What is the relationship between [character X] and [character Y]?
- What is the setting of the story?
- What is the significance of [object X] on the rest of the story?
- How is [theme X] explored throughout the story?
- Story-specific questions

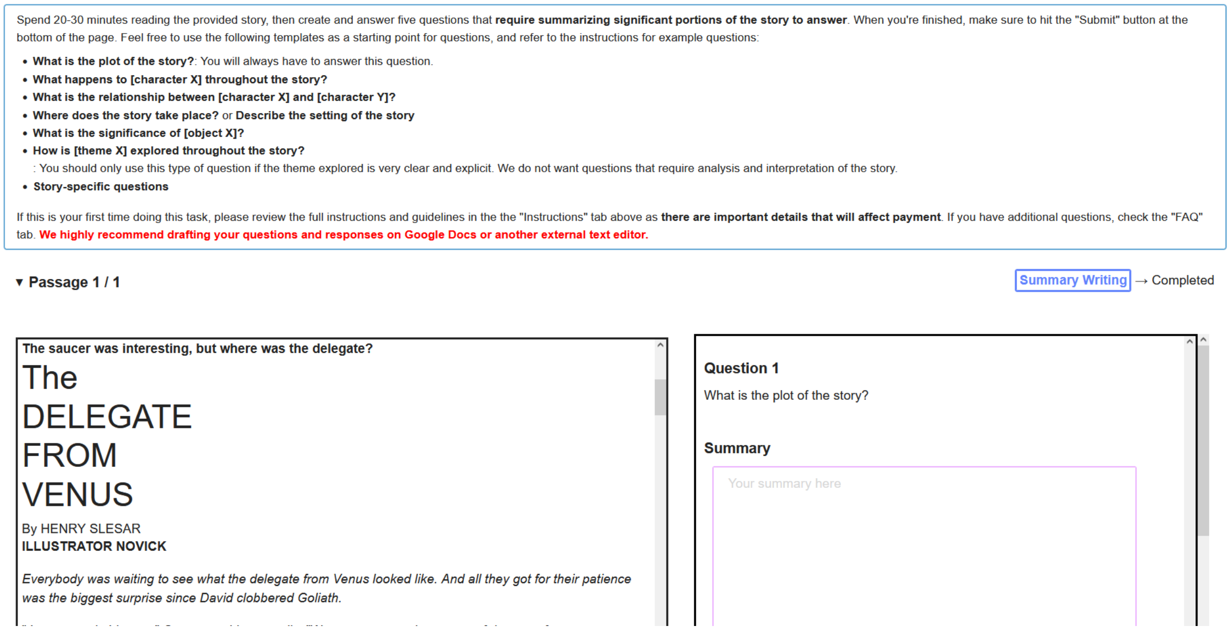
Writers always answer the question “What is the plot of the story?”. For more subjective templates such as “What is the significance of [object X]?” or “How is [theme X] explored?”, we ask the writers to use these templates only in cases where they believe the answer will be clear and unambiguous to someone who has read the story carefully.

A.4.1.2 Crowdsourcing Interfaces

We show screenshots of our UIs and abbreviated task instructions for writing and reviewing summaries in Figures A.5 and A.6, respectively.

A.4.1.3 Comparing Upwork and Undergraduates

Generally, we found that both Upwork and undergraduate workers took the task seriously and produced quality summaries. Writers from Upwork qualitatively produced slightly higher quality responses, perhaps because we were able to filter more aggressively for relevant backgrounds and skills when hiring on Upwork. Hiring writers on Upwork was more expensive than hiring student writers.



Spend 20-30 minutes reading the provided story, then create and answer five questions that **require summarizing significant portions of the story to answer**. When you're finished, make sure to hit the "Submit" button at the bottom of the page. Feel free to use the following templates as a starting point for questions, and refer to the instructions for example questions:

- **What is the plot of the story?**: You will always have to answer this question.
- **What happens to [character X] throughout the story?**
- **What is the relationship between [character X] and [character Y]?**
- **Where does the story take place?** or **Describe the setting of the story**
- **What is the significance of [object X]?**
- **How is [theme X] explored throughout the story?**
  : You should only use this type of question if the theme explored is very clear and explicit. We do not want questions that require analysis and interpretation of the story.
- **Story-specific questions**

If this is your first time doing this task, please review the full instructions and guidelines in the the "Instructions" tab above as **there are important details that will affect payment**. If you have additional questions, check the "FAQ" tab. **We highly recommend drafting your questions and responses on Google Docs or another external text editor.**

The saucer was interesting, but where was the delegate?

The
DELEGATE
FROM
VENUS

By HENRY SLESAR
ILLUSTRATOR NOVICK

Everybody was waiting to see what the delegate from Venus looked like. And all they got for their patience was the biggest surprise since David clobbered Goliath.

Question 1

What is the plot of the story?

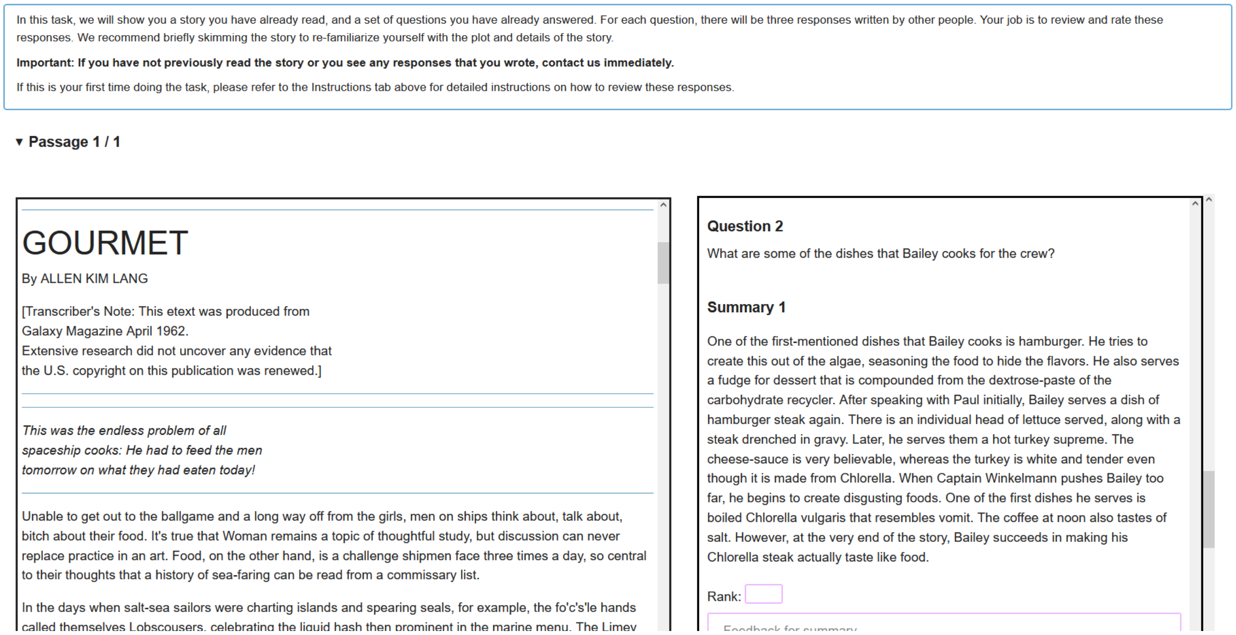
Summary

Figure A.5: Screenshot of the writing UI. Workers are shown the story on the left and five questions on the right, and they are tasked with writing responses to each of the questions. If the worker is the first person to work on a story, they write four questions about the story to answer (The question "What is the plot?" is always asked), and we provide the worker with a list of question templates in the UI to help them write good questions.

Anecdotally, the workers we hired from both populations enjoyed the tasks, and we see this as a significant advantage to using popular fiction in benchmark tasks. However, we did find that some Upwork contractors quit our task during the course of data collection, and some mentioned that our task paid less than other tasks on Upwork. Because students were hired for long-term contracts (on the order of months), they did not drop out of the data collection process, but working with them did require careful work scheduling around exams and breaks.

A.4.2 Dataset Examples

Table A.6 shows the full references for the example in Table 5.1. Table A.7 shows additional examples from SQuALITY.



In this task, we will show you a story you have already read, and a set of questions you have already answered. For each question, there will be three responses written by other people. Your job is to review and rate these responses. We recommend briefly skimming the story to re-familiarize yourself with the plot and details of the story.

Important: If you have not previously read the story or you see any responses that you wrote, contact us immediately.

If this is your first time doing the task, please refer to the Instructions tab above for detailed instructions on how to review these responses.

▾ Passage 1 / 1

GOURMET

By ALLEN KIM LANG

[Transcriber's Note: This etext was produced from
Galaxy Magazine April 1962.
Extensive research did not uncover any evidence that
the U.S. copyright on this publication was renewed.]

*This was the endless problem of all
spaceship cooks: He had to feed the men
tomorrow on what they had eaten today!*

Unable to get out to the ballgame and a long way off from the girls, men on ships think about, talk about, bitch about their food. It's true that Woman remains a topic of thoughtful study, but discussion can never replace practice in an art. Food, on the other hand, is a challenge shipmen face three times a day, so central to their thoughts that a history of sea-faring can be read from a commissary list.

In the days when salt-sea sailors were charting islands and spearing seals, for example, the fo'c's'le hands called themselves Lobscousers, celebrating the liquid hash then prominent in the marine menu. The Limey

Question 2

What are some of the dishes that Bailey cooks for the crew?

Summary 1

One of the first-mentioned dishes that Bailey cooks is hamburger. He tries to create this out of the algae, seasoning the food to hide the flavors. He also serves a fudge for dessert that is compounded from the dextrose-paste of the carbohydrate recycler. After speaking with Paul initially, Bailey serves a dish of hamburger steak again. There is an individual head of lettuce served, along with a steak drenched in gravy. Later, he serves them a hot turkey supreme. The cheese-sauce is very believable, whereas the turkey is white and tender even though it is made from Chlorella. When Captain Winkelmann pushes Bailey too far, he begins to create disgusting foods. One of the first dishes he serves is boiled Chlorella vulgaris that resembles vomit. The coffee at noon also tastes of salt. However, at the very end of the story, Bailey succeeds in making his Chlorella steak actually taste like food.

Rank:

Feedback for summary

Figure A.6: Screenshot of the reviewing UI. Workers are shown the story on the left and five questions on the right. Each of the questions has three responses that the worker is tasked with ranking from best to worst. Additionally, for each response, the worker is instructed to highlight typos and factual errors, as well as provide written feedback to the writer. This feedback is later provided to the writer to help them improve their responses in subsequent rounds of writing.

A.4.3 Training Details

We train models for 5 epochs with the AdamW optimizer and a linear decay with warmup learning rate schedule. Because of the relatively small size of the training data, we focus on tuning regularization parameters when training the models. We tune the initial learning rate, warmup ratio, weight decay, and label smoothing with grid search over a range of values for each hyperparameter. Models were selected based on the loss on the validation dataset and the ability to generate fluent summaries on the validation dataset. We present the search space for each parameter and the optimal model configurations for each model in Table A.8. Our experiments with PEGASUS predominantly led to models that produced degenerate summaries consisting of a single sentence repeated. The final model we use is from the official Google-internal implementation courtesy of the original authors. LED models were trained on a single Nvidia Quadro RTX 8000. Other models were trained on a single Nvidia V100.

A.4.4 Model Outputs

We present sample model outputs in Table A.9.

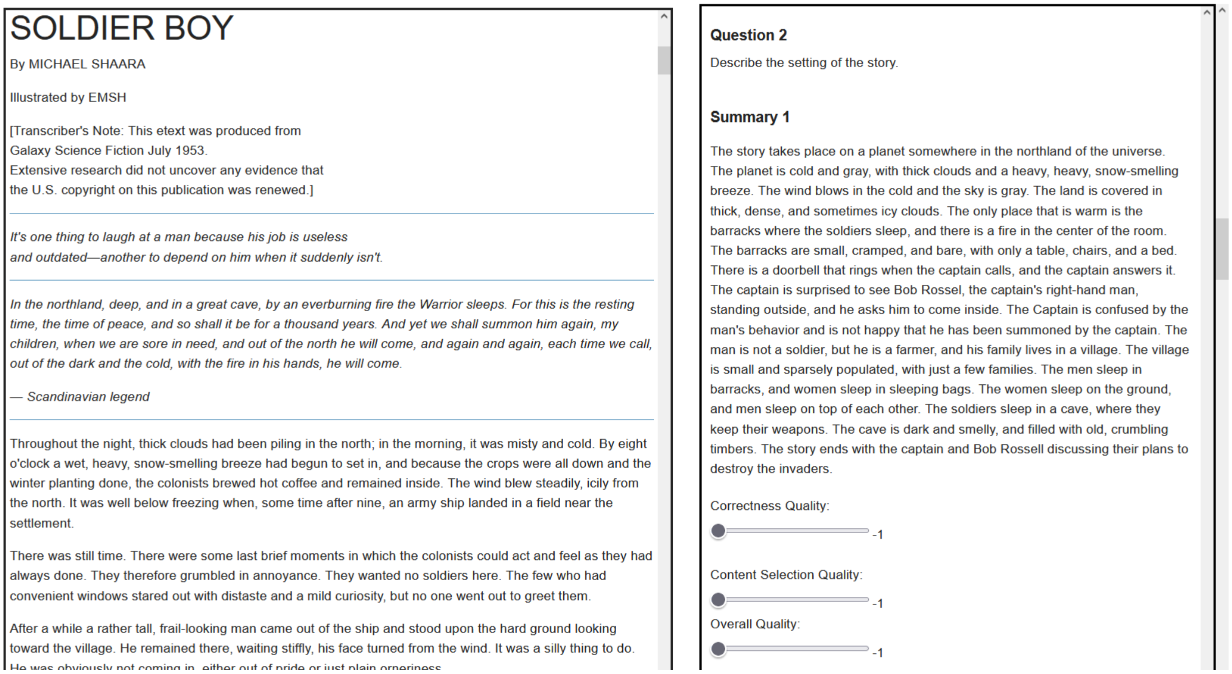
A.4.5 Human Evaluation

As the task is labor-intensive, we use four of the same Upwork writers for the human evaluation as for the data collection. Workers may have previously read the story and thus answered the questions, and we are careful to not show workers their own responses. If they have not previously read the stories, workers are paid to read the story. Workers are informed that the responses are a mixture of human- and machine-written, but not informed which responses are which. We pay workers $8/task and an additional $8 if they have not previously read the story. All workers complete the same number of tasks.

We ask human raters to (re-)read the story, and then evaluate the quality of summaries along three axes:

- Correctness: Presence of factual errors in responses, where a factual error is a statement that contradicts the story, or is not directly stated, heavily implied, or logically entailed by the story.
- Coverage: The degree to which the response contains all information and details from the story that are relevant to answering the question.
- Overall: Overall quality of the response, the primary considerations of which are the readability/intelligibility of the response, the correctness, and the coverage. We ask raters to use their best judgment in balancing these factors, as well as to incorporate other factors such as conciseness, repetitiveness, and copying.

We show the standard deviation of property ratings across questions in Table A.10.



SOLDIER BOY

By MICHAEL SHAARA

Illustrated by EMSH

[Transcriber's Note: This etext was produced from
Galaxy Science Fiction July 1953.
Extensive research did not uncover any evidence that
the U.S. copyright on this publication was renewed.]

It's one thing to laugh at a man because his job is useless and outdated—another to depend on him when it suddenly isn't.

In the northland, deep, and in a great cave, by an everburning fire the Warrior sleeps. For this is the resting time, the time of peace, and so shall it be for a thousand years. And yet we shall summon him again, my children, when we are sore in need, and out of the north he will come, and again and again, each time we call, out of the dark and the cold, with the fire in his hands, he will come.

— Scandinavian legend

Throughout the night, thick clouds had been piling in the north; in the morning, it was misty and cold. By eight o'clock a wet, heavy, snow-smelling breeze had begun to set in, and because the crops were all down and the winter planting done, the colonists brewed hot coffee and remained inside. The wind blew steadily, icily from the north. It was well below freezing when, some time after nine, an army ship landed in a field near the settlement.

There was still time. There were some last brief moments in which the colonists could act and feel as they had always done. They therefore grumbled in annoyance. They wanted no soldiers here. The few who had convenient windows stared out with distaste and a mild curiosity, but no one went out to greet them.

After a while a rather tall, frail-looking man came out of the ship and stood upon the hard ground looking toward the village. He remained there, waiting stiffly, his face turned from the wind. It was a silly thing to do. He was obviously not coming in, either out of pride or just plain orneriness

Question 2

Describe the setting of the story.

Summary 1

The story takes place on a planet somewhere in the northland of the universe. The planet is cold and gray, with thick clouds and a heavy, heavy, snow-smelling breeze. The wind blows in the cold and the sky is gray. The land is covered in thick, dense, and sometimes icy clouds. The only place that is warm is the barracks where the soldiers sleep, and there is a fire in the center of the room. The barracks are small, cramped, and bare, with only a table, chairs, and a bed. There is a doorbell that rings when the captain calls, and the captain answers it. The captain is surprised to see Bob Rossel, the captain's right-hand man, standing outside, and he asks him to come inside. The Captain is confused by the man's behavior and is not happy that he has been summoned by the captain. The man is not a soldier, but he is a farmer, and his family lives in a village. The village is small and sparsely populated, with just a few families. The men sleep in barracks, and women sleep in sleeping bags. The women sleep on the ground, and men sleep on top of each other. The soldiers sleep in a cave, where they keep their weapons. The cave is dark and smelly, and filled with old, crumbling timbers. The story ends with the captain and Bob Rossell discussing their plans to destroy the invaders.

Correctness Quality: -1

Content Selection Quality: -1

Overall Quality: -1

Figure A.7: Screenshot of the human evaluation UI. Workers are shown the story on the left and five questions on the right. Each of the questions has three responses. For each response, the worker is instructed to rate the responses along the properties of correctness, coverage, and overall quality each along a scale of 1–100. Because the worker is shown three responses at a time, their ratings of each response induce a ranking over the responses. Additionally, workers are asked to highlight errors in responses in order to help them decide on the correctness property.

Title: Pick A Crime (https://www.gutenberg.org/ebooks/51656)	
Q: What is the CPA and what does it do?	
The Crime Prevention Association is an organization that stops crime. Instead of capturing criminals, the goal of the Association is to prevent the crime from ever happening. They implement thousands of crime-prevention methods and devices. There are many amateur cops who constantly follow criminals around in hopes of catching them in the act so that they may be hailed a hero and given a promotion. Hendricks even explains that the kids have junior CPA clubs, where they record the criminals in little cardboard boxes. They will also follow the criminals around until they die. There are millions of microphones hidden by the CPA everywhere, and any threatening messages are sent to the CPA Brain. The CPA Brain is a monster electronic calculator that can alert police helicopters of any threatening messages, and there are also many hidden TVs and metal detectors. For Arson, heat detectors exist too, and chemical poisoning has made it impossible for people to get poisoned. There are shock treatments, encephalographic devices, a form of prefrontal lobotomy, and a dozen other treatments to reform criminals.	The CPA, Crime Prevention Association, is a system that detects different kinds of crimes and prevents them from happening. Thousands of robots and devices make crimes impossible. The association will not punish any crime, instead, the criminal will be send to a CPA hospital for some treatments that will result in getting the best jobs. The CPA also hands out ID cards that states one's tendency to commit crimes. The CPA has robot bartenders that can detect the drunkenness of a person and prevent anyone from actually getting drunk. There is WSDA teaching judo and jujitsu to women. There are spy cameras and speakers in each alley and street watching every person all the time to prevent all kinds of crimes. The CPA Brain will catch sentences that indicate crimes and watch them more carefully. There are heat-detectors, gun and knife detector, chemical detectors, etc. The CPA brainwashes people, making them believe that crimes are filthy. The treatment will make the criminal's brain catch every attempt that he or she tries to commit a crime and prevents it from happening.
The CPA is Crime Prevention Organization. It fights crime by all means and reduces its rates to a very small level. They put microphones and detectors everywhere to hear the conspiracies. They place robots as bartenders to control the level of alcohol in visitors to prevent them being drunk. They make all the women learn self-defense. The organization's made crime almost impossible and they do not punish for it, but prevent. All who tried to commit a crime are given free treatment. The CPA hospitals treat those few criminals for free and make them unable to commit any further crime. CPA seems to be everywhere, those who tell about the crime are highly rewarded. Neon signs, TV, radio and other means constantly remind people that crime is filth.	The CPA is meant to prevent crime and not punish crime. It stands for Crime Prevention Association. The CPA organization has made crime nearly impossible through various methods of surveillance and intelligence gathering. The crime was not punished by the CPA but addressed by sending the person to a hospital for expensive treatment to correct and remove the deviance from the person's mind. A CPA ID card is required to be carried by everyone and when asked, a person has to present the ID card. Being drunk is illegal according to the rules of the CPA.

Table A.6: The four full human-written references from Table 5.1.

Title: Tolliver's Orbit (https://www.gutenberg.org/ebooks/61053)	
Q: Describe the equipment used throughout the story.	
Tolliver is a pilot, but while at the Ganymede branch he drives a tractor. One of the equipment used during the story is the automatic flight. An automatic flight allows loaded ships to take a slow and economical orbit using automatic signaling equipment towards Earth. As the loaded ship gets closer to Earth, it is boarded by pilots that land the ship. Another piece of equipment mentioned are spacesuits. The spacesuits involve valves and seals and microphones for people to communicate with each other in the spacesuits. The communication is activated by a switch under the chin on the helmet of the spacesuit. They also come with a heavy knife.	Various types of transportation are used throughout the story - tractors to travel on Ganymede between the city and the spaceport, spaceships requiring a lot of fuel and economy orbits which require less fuel but take much longer to get to the place. In a storeroom there are plenty spacesuits, some of which need replacement. Knives are standard suit equipment. Spaceships are equipped with airlocks, ladders and switch-cover. In the control room there is an acceleration seat, a button to set off, a radio and TV, with a screen to see the other side of the call.
Tolliver is first assigned to use an airtight tractor to transport to and from the spaceport. This tractor is like a regular one, but built specifically to trek across Ganymede with its gravity. When Tolliver and Betty are locked into Jeffers' office, he uses a lighter and paper to bend the plastic of the door. Then, he uses a knife to cut through the plastic of the dome. Finally, Tolliver and Betty board a ship, where the orbit is automatically preset in order to preserve fuel. The ship, which Tolliver knows how to operate, is airlocked. Betty uses a transmitter to contact Space Patrol.	Firstly, Tolliver takes Betty towards Jeffers' office on a tractor since it can go through the frozen surface of Ganymede. Then later, when Betty and Tolliver were put in the empty office, Tolliver uses a lighter to light up the mess of discarded records so that the plastic can be bent. Later, inside the storage room, Tolliver finds some spacesuits for the two to wear. Then finally, when they gets to the control room, they gets onto the acceleration seat. Using the ship, the two fly into the economy orbit for Earth in order to escape. In the end, Betty uses the scanner and microphone to make a call to the Space Patrol so that they will arrest Jeffers.
Title: Gourmet (https://www.gutenberg.org/ebooks/51597)	
Q: What are some of the dishes that Bailey cooks for the crew?	
The dishes Bailey cooks for the crew varies greatly, ranging from artificial vegetables to mock-meats. One dish that he makes is a mock-meat hamburger, with the pressed Chlorella tinted pink and seasoned by oregano and thyme. The dish is accompanied by dessert - a fudge made from dextrose-paste. More mock-meat dishes include a hamburger steak covered in a rich, meaty gravy lavishly seasoned with garlic. Another dish includes a mock individual head of lettuce dressed with vinegar and oil. The lettuce was made by Bailey constructing each synthetic lettuce leaf, with the narrator guessing the process to be out of pressing, rolling and shaping a green Chlorella paste. In contrast to some of the delicious dishes that Bailey makes, the Cook also delivers some less tasty meals in response to the Captain's critiques. These included boiled Chlorella vulgaris in some soup and subpar algaeburgers. Bailey's final dish in the story - and the best one yet - is an artificial steak that greets the crew with a barbecue smell. It is drenched with gravy and seasoned with a peppery and garlicy taste, and as the crew eats it, they find that the usually pond-scum taste that accompanies each repurposed chlorella meal is gone and instead, the taste and texture reflects actual steak.	One of the first-mentioned dishes that Bailey cooks is hamburger. He tries to create this out of the algae, seasoning the food to hide the flavors. He also serves a fudge for dessert that is compounded from the dextrose-paste of the carbohydrate recycler. After speaking with Paul initially, Bailey serves a dish of hamburger steak again. There is an individual head of lettuce served, along with a steak drenched in gravy. Later, he serves them a hot turkey supreme. The cheese-sauce is very believable, whereas the turkey is white and tender even though it is made from Chlorella. When Captain Winkelmann pushes Bailey too far, he begins to create disgusting foods. One of the first dishes he serves is boiled Chlorella vulgaris that resembles vomit. The coffee at noon also tastes of salt. However, at the very end of the story, Bailey succeeds in making his Chlorella steak actually taste like food.
Throughout their trip, Bailey does the best he can in order to replicate traditional food using the Algae. To impress the Captain, Bailey cooks a wide variety of foods including algae burgers, fudge, Steak with gravy and a head of lettuce, Hot turkey with cornbread and butter sauce, and medium rare steak. None of these foods impressed the Captain, so Bailey went back to cooking unappealing food such as a porridge-like broth and bad coffee. At the end, Bailey serves a new type of steak, which is hinted to be human steak from the Captain.	Bailey made a lot of different dishes while working on the Sale ship. He cooked a hamburger and a fudge. He made a steak with rich meat gravy and lettuce, vinegar, and oil. An ersatz hot turkey supreme with a cheese sauce, cornbread, and a pottage was also served at some point. All of these were criticized by Captain Winkelmann. Mostly Bailey was working on the taste of steak, which at the end of the story, he managed to perfect to a certain extent, partly thanks to the captain's constant remarks.

Table A.7: Additional example questions and reference summaries from SQuALITY.

Hyperparameter	LR	WR	WD	LS
Search space	{1e-5, 3e-5, 5e-5, 1e-4}	{0.0, 0.1, 0.2}	{0.0, 0.01, 0.1}	{0.0, 0.1, 0.2}
LED	1e-4	0.1	0.01	0.1
PEGASUS	-	-	-	-
BART	5e-5	0.1	0.01	0.1
BART+DPR	5e-5	0.1	0.01	0.1

Table A.8: (Top) Search space for the initial learning rate (LR), warmup ratio (WR), weight decay (WD), label smoothing (LS). (Bottom) Optimal hyperparameter configurations for models. The final PEGASUS model we use is from the official Google-internal implementation courtesy of the original authors.

<table>
<tr><td colspan="2">Title: Retief of the Red-Tape Mountain (https://www.gutenberg.org/ebooks/61146)</td></tr>
<tr><td colspan="2">Q: What is the relationship between the Jaqs and the Terrestrials throughout the story?</td></tr>
<tr><td><b>Reference</b>: The Jaqs and the Terrestrials fight each other throughout the story. It started when a human saw a Jaq and thought it was some type of native game and shot it. From that incident, the Jaqs concluded that the humans were sportsmen like themselves and responded by going to one of the farms and killing two cows. Since then, the two sides have been attacking back and forth, and the humans think the Jaqs are fighting against them...</td><td><b>LED</b>: Retief is a vice-consul at the Embassy of the Mosaic of the Two Dawns. He is in charge of the affairs of the Embassy and is responsible for keeping the diplomatic relations between the two planets in check. He is also responsible for keeping the diplomatic relations in check by sending Retief on his expeditious trip to the planet Adobe. When Retief arrives at the planet, he is greeted by a large Flap-jack, a creature with talons that look like lobsters...</td></tr>
<tr><td><b>BART</b>: The Terrestrials and the Jaqs have a tense relationship throughout the story. The Terrans have attempted to establish contact with the native life form, the Jaq, in order to try to gain their trust and gain information about their native life forms. The Jaqs are hostile to the Terrans because they consider them to be an invasive species that are trying to take over their home planet, which they consider to be uninhabited. The Jaqs have a history of war with the Terran settlers...</td><td><b>BART+DPR</b>: The Terrestrials and the Flap-jacks are an alien race that live on the planet Adoban. They are hostile to humans and have attempted to stir up trouble with an intelligent alien life form, the Jaq, three months ago. The humans are attempting to establish trade with the aliens in order to gain access to the planet's resources, but the aliens are having none of it. They have no intention of trading with the humans and are only interested in trading with them for food and...</td></tr>
</table>

Table A.9: Example model generations on SQuALITY.

Model	Correctness	Coverage	Overall
BART	$34.8_{16.9}$	$15.6_{13.3}$	$18.1_{13.1}$
BART+DPR	$45.4_{15.8}$	$24.3_{15.7}$	$27.9_{16.5}$
Human	$94.1_{5.5}$	$88.8_{10.9}$	$91.3_{7.7}$

Table A.10: Human evaluation results for two models and a human-written response. Ratings for each property are averaged across 3 workers, then averaged across questions. Standard deviation of property ratings across questions are shown in underscore.

Bibliography

Ekaterina Ageeva, Francis M. Tyers, Mikel L. Forcada, and Juan Antonio Pérez-Ortiz. 2015. Evaluating machine translation for assimilation via a gap-filling task. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*, Antalya, Turkey. European Association for Machine Translation.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint 2108.07732*.

Stephen H. Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alexander Ratner, Braden Hancock, Houman Alborzi, Rahul Kuchhal, Christopher Ré, and Rob Malkin. 2018. Snorkel drybell: A case study in deploying weak supervision at industrial scale. In *SIGMOD*. ACM.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Roy Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second PASCAL recognising textual entailment challenge.

Max Bartolo, Tristan Thrush, Sebastian Riedel, Pontus Stenetorp, Robin Jia, and Douwe Kiela. 2021. Models in the loop: Aiding crowdworkers with generative annotation assistants. *arXiv preprint 2112.09062*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint 2004.05150*.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *TAC*.

BIG-bench collaboration. 2021. Beyond the imitation game: Measuring and extrapolating the capabilities of language models. *In preparation*.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. Results of the WMT16 metrics shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 199–231, Berlin, Germany. Association for Computational Linguistics.

Samuel 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.

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. Association for Computational Linguistics.

Sven Buechel, Anneke Buffone, Barry Slaff, Lyle Ungar, and João Sedoc. 2018. Modeling empathy and distress in reaction to news stories. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluation the role of bleu in machine translation research. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Yllias Chali and Maheedhar Kolla. 2004. Summarization techniques at duc 2004. In *In Proceedings of the Document Understanding Conference*. Citeseer.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint 1312.3005*.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686.

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

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018b. Ultra-fine entity typing. In *Proceedings of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics.

Prafulla Kumar Choubey, Jesse Vig, Wenhao Liu, and Nazneen Fatema Rajani. 2021. MoFE: Mixture of factual experts for controlling hallucinations in abstractive summarization. *arXiv preprint 2110.07166*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019a. Boolq: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019b. BAM! Born-again multi-task networks for natural language understanding. In *Proceedings of the Association of Computational Linguistics (ACL)*. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. 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 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *LREC 2018*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Josef Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, Steve Pulman, Ted Briscoe, Holger Maier, and Karsten Konrad. 1996. Using the framework. Technical report, The FraCaS Consortium.

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 (NeurIPS)*. Curran Associates, Inc.

Hal Daume III and Daniel Marcu. 2005. Bayesian summarization at duc and a suggestion for extrinsic evaluation. In *Proceedings of the Document Understanding Conference, DUC-2005, Vancouver, USA*.

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The CommitmentBank: Investigating projection in naturally occurring discourse. To appear in *Proceedings of Sinn und Bedeutung 23*. Data can be found at https://github.com/mcdm/CommitmentBank/.

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.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of IWP*.

Bonnie Dorr, Christof Monz, Douglas Oard, David Zajic, and Richard Schwartz. 2004. Extrinsic evaluation of automatic metrics for summarization. Technical report, MARYLAND UNIV COLLEGE PARK INST FOR ADVANCED COMPUTER STUDIES.

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.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. On the origin of hallucinations in conversational models: Is it the datasets or the models? *arXiv preprint 2204.07931*.

Allyson Ettinger, Sudha Rao, Hal Daumé III, and Emily M Bender. 2017. Towards linguistically generalizable NLP systems: A workshop and shared task. In *First Workshop on Building Linguistically Generalizable NLP Systems*.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. 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 3938–3948.

Alexander R Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2021. QAFactEval: Improved QA-based factual consistency evaluation for summarization. *arXiv preprint 2112.08542*.

Tobias Falke, Leonardo FR Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 2214–2220.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.

Manaal Faruqui and Dipanjan Das. 2018. Identifying well-formed natural language questions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Katja Filippova. 2020. Controlled hallucinations: Learning to generate faithfully from noisy data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 864–870.

Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. *International Conference on Machine Learning (ICML)*.

Saadia Gabriel, Asli Celikyilmaz, Rahul Jha, Yejin Choi, and Jianfeng Gao. 2021. Go figure: A meta evaluation of factuality in summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 478–487.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software*.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2022. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *arXiv preprint 2202.06935*.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9. Association for Computational Linguistics.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. 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 609–614, Minneapolis, Minnesota. Association for Computational Linguistics.

Ben Goodrich, Vinay Rao, Peter J. Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 166–175, New York, NY, USA. ACM.

Jan Gorodkin. 2004. Comparing two k-category assignments by a k-category correlation coefficient. *Comput. Biol. Chem.*, 28(5-6):367–374.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2016. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of EMNLP 2017*.

Hiroaki Hayashi, Prashant Budania, Peng Wang, Chris Ackerson, Raj Neervannan, and Graham Neubig. 2021. WikiAsp: A dataset for multi-domain aspect-based summarization. *Transactions of the Association for Computational Linguistics*, 9:211–225.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint 2103.03874*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *Proceedings of NAACL 2016*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint 1503.02531*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. True: Re-evaluating factual consistency evaluation. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 161–175.

David M Howcroft, Anja Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A Hasan, Saad Mahamood, Simon Mille, Emiel Van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Neslihan Iskender, Tim Polzehl, and Sebastian Möller. 2021. Reliability of human evaluation for text summarization: Lessons learned and challenges ahead. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 86–96.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Chris Kedzie, Kathleen McKeown, and Hal Daume III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Association for Computational Linguistics.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. 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.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *ICLR 2015*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-Thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz. 2019. A surprisingly robust trick for the Winograd schema challenge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4837–4842, Florence, Italy. Association for Computational Linguistics.

Kalpesh Krishna and Mohit Iyyer. 2019. Generating question-answer hierarchies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2321–2334, Florence, Italy. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. 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 540–551, Hong Kong, China. Association for Computational Linguistics.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346.

Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev.

2021. BookSum: A collection of datasets for long-form narrative summarization. *arXiv preprint 2105.08209*.

Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. 2020. AQuaMuSe: Automatically generating datasets for query-based multi-document summarization. *arXiv preprint 2010.12694*.

Philippe Laban, Tobias Schnabel, Paul N Bennett, and Marti A Hearst. 2021. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *arXiv preprint 2111.09525*.

Faisal Ladhak, Bryan Li, Yaser Al-Onaizan, and Kathleen McKeown. 2020. Exploring content selection in summarization of novel chapters. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5043–5054, Online. Association for Computational Linguistics.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231. Association for Computational Linguistics.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2020. FlauBERT: Unsupervised language model pre-training for french. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2479–2490.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Analyzing sentence fusion in abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 104–110.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

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.

Linjie Li, Jie Lei, Zhe Gan, Licheng Yu, Yen-Chun Chen, Rohit Pillai, Yu Cheng, Luowei Zhou, Xin Eric Wang, William Yang Wang, et al. 2021. VALUE: A multi-task benchmark for video-and-language understanding evaluation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Margaret Li and Julian Michael. 2022. Overconfidence in the face of ambiguity with adversarial data. In *Proceedings of the First Workshop on Dynamic Adversarial Data Collection*, pages 30–40.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, et al. 2020. XGLUE: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Alisa Liu, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2022. WANLI: Worker and ai collaboration for natural language inference dataset creation. *arXiv preprint 2201.05955*.

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*. Association for Computational Linguistics.

Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, et al. 2021. GLGE: A new general language generation evaluation benchmark. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 408–420.

Feifan Liu and Yang Liu. 2009. Exploring correlation between ROUGE and human evaluation on meeting summaries. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(1):187–196.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Association for Computational Linguistics.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019b. Inoculation by fine-tuning: A method for analyzing challenge datasets. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019c. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. *arXiv preprint 1904.09482*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019d. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

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

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2020. Gender bias in neural natural language processing. In *Logic, Language, and Security*, pages 189–202. Springer.

Inderjeet Mani, David House, Gary Klein, Lynette Hirschman, Therese Firmin, and Beth M Sundheim. 1999. The tipster summac text summarization evaluation. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics.*

Brian W Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint 1806.08730.*

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics.

Richard T. McCoy and Tal Linzen. 2019. Non-entailed subsequences as a challenge for natural language inference. In *Proceedings of the Society for Computational in Linguistics (SCiL) 2019*.

Rui Meng, Khushboo Thaker, Lei Zhang, Yue Dong, Xingdi Yuan, Tong Wang, and Daqing He. 2021. Bringing structure into summaries: a faceted summarization dataset for long scientific documents. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1080–1089, Online. Association for Computational Linguistics.

Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. Parlai: A dialog research software platform. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–84.

George A Miller. 1995. WordNet: a lexical database for english. *Communications of the ACM*.

Aakanksha Naik, Abhilasha Ravichander, Norman M. Sadeh, Carolyn Penstein Rosé, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *International Conference on Computational Linguistics (COLING)*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Feng Nan, Cicero dos Santos, Henghui Zhu, Patrick Ng, Kathleen Mckeown, Ramesh Nallapati, Dejiao Zhang, Zhiguo Wang, Andrew O Arnold, and Bing Xiang. 2021a. Improving factual consistency of abstractive summarization via question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6881–6894.

Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero Nogueira dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen McKeown, and Bing Xiang. 2021b. Entity-level factual consistency of abstractive text summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2727–2733, Online. Association for Computational Linguistics.

Nikita Nangia and Samuel R. Bowman. 2019. Human vs. Muppet: A conservative estimate of human performance on the GLUE benchmark. In *Proceedings of the Association of Computational Linguistics (ACL)*. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1063–1072, Vancouver, Canada. Association for Computational Linguistics.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the human language technology conference of the north american chapter of the association for computational linguistics: Hlt-naacl 2004*, pages 145–152.

Allen Nie, Erin D Bennett, and Noah D Goodman. 2017. Dissent: Sentence representation learning from explicit discourse relations. *arXiv preprint 1710.04334*.

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. *NAACL HLT 2019*, page 48.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics.

Richard Yuanzhe Pang, Adam Lelkes, Vinh Tran, and Cong Yu. 2021a. AgreeSum: Agreement-oriented multi-document summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3377–3391, Online. Association for Computational Linguistics.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, et al. 2021b. QuALITY: Question answering with long input texts, yes! *arXiv preprint 2112.08608*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Ji Yoon Han, Jangwon Park, Chisung Song, Junseong Kim, Youngsook Song, Taehwan Oh, et al. 2021. KLUE: Korean language understanding evaluation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

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.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL 2018*.

Jason Phang, Angelica Chen, William Huang, and Samuel R Bowman. 2021. Adversarially constructed evaluation sets are more challenging, but may not be fair. *arXiv preprint 2111.08181*.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on STILTs: Supplementary training on intermediate labeled-data tasks. *arXiv preprint 1811.01088*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: The word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Association for Computational Linguistics.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018a. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018b. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191.

Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for MT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 751–762.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Unpublished ms. available through a link at `https://blog.openai.com/language-unsupervised/`.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

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. *Journal of Machine Learning Research*, 21:1–67.

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. Association for Computational Linguistics.

Hannah Rashkin, David Reitter, Gaurav Singh Tomar, and Dipanjan Das. 2021. Increasing faithfulness in knowledge-grounded dialogue with controllable features. 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 704–718.

Tim Rocktäschel, Edward Grefenstette, Moritz Hermann, Karl, Tomáš Kočiský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2017. Sluice networks: Learning what to share between loosely related tasks. *arXiv preprint 1705.08142*.

Rachel Rudinger, Chandler May, and Benjamin Van Durme. 2017. Social bias in elicited natural language inferences. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 74–79.

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*. Association for Computational Linguistics.

Piotr Rybak, Robert Mroczkowski, Janusz Tracz, and Ireneusz Gawlik. 2020. KLEJ: Comprehensive benchmark for polish language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1191–1201.

Keisuke Sakaguchi and Benjamin Van Durme. 2018. Efficient online scalar annotation with bounded support. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 208–218, Melbourne, Australia. Association for Computational Linguistics.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. SocialIQa: Commonsense reasoning about social interactions. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. *arXiv preprint 2206.05802*.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352.

Natalie Schluter. 2017. The limits of automatic summarisation according to ROUGE. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45, Valencia, Spain. Association for Computational Linguistics.

Nathan Schneider and Noah A Smith. 2015. A corpus and model integrating multiword expressions and supersenses. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Association for Computational Linguistics.

Karin Kipper Schuler. 2005. *Verbnet: A Broad-coverage, Comprehensive Verb Lexicon*. Ph.D. thesis.

Carson T Schütze. 1996. *The empirical base of linguistics: Grammaticality judgments and linguistic methodology*. University of Chicago Press.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Li 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 CoNLL*.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020a. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892.

Thibault Sellam, Amy Pu, Hyung Won Chung, Sebastian Gehrmann, Qijun Tan, Markus Freitag, Dipanjan Das, and Ankur Parikh. 2020b. Learning to evaluate translation beyond english: BLEURT submissions to the WMT metrics 2020 shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 921–927.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. In *ICLR 2017*.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, et al. 2022. SCROLLS: Standardized comparison over long language sequences. *arXiv preprint 2201.03533*.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

Eric Smith, Orion Hsu, Rebecca Qian, Stephen Roller, Y-Lan Boureau, and Jason Weston. 2022. Human evaluation of conversations is an open problem: comparing the sensitivity of various methods for evaluating dialogue agents. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 77–97.

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.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 231–235.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.

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*, 33:3008–3021.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J. Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. In *Proceedings of ICLR*.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint 2107.02137*.

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.

Priyam Tejaswin, Dhruv Naik, and Pengfei Liu. 2021. How well do you know your summarization datasets? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3436–3449, Online. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *International Conference on Learning Representations (ICLR)*.

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

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.

Harsh Trivedi, Heeyoung Kwon, Tushar Khot, Ashish Sabharwal, and Niranjan Balasubramanian. 2019. Repurposing entailment for multi-hop question answering tasks. 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 2948–2958, Minneapolis, Minnesota. Association for Computational Linguistics.

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).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Jesse Vig, Alexander R Fabbri, and Wojciech Kryściński. 2021. Exploring neural models for query-focused summarization. *arXiv preprint 2112.07637*.

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

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R Bowman. 2022. SQuALITY: Building a long-document summarization dataset the hard way. *arXiv preprint 2205.11465*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Alex Wang, Ian F. Tenney, Yada Pruksachatkun, Katherin Yu, Jan Hula, Patrick Xia, Raghu Pappagari, Shuning Jin, R. Thomas McCoy, Roma Patel, Yinghui Huang, Jason Phang, Edouard Grave, Haokun Liu, Najoung Kim, Phu Mon Htut, Thibault F'evry, Berlin Chen, Nikita Nangia, Anhad Mohananey, Katharina Kann, Shikha Bordia, Nicolas Patry, David Benton, Ellie Pavlick, and Samuel R. Bowman. 2019c. `jiant` 1.2: A software toolkit for research on general-purpose text understanding models. `http://jiant.info/`.

Alex Wang and Thomas Wolf. 2020. Overview of the sustaiNLP 2020 shared task. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 174–178.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics (TACL)*.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 996–1005.

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.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiaohong Li, Zhi Yuan Lim, Sidik Soleman, Rahmad Mahendra, Pascale Fung, Syafri Bahar, et al. 2020. IndoNLU: Benchmark and resources for evaluating indonesian natural language understanding. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 843–857.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Association for Computational Linguistics.

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.

Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback. *arXiv preprint 2109.10862*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. In *ICLR 2017*.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, et al. 2020. CLUE: A chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772.

Liang Xu, Xiaojing Lu, Chenyang Yuan, Xuanwei Zhang, Huilin Xu, Hu Yuan, Guoao Wei, Xiang Pan, Xin Tian, Libo Qin, et al. 2021. FewCLUE: A chinese few-shot learning evaluation benchmark. *arXiv preprint 2107.07498*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems (NeurIPS)*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34.

Fabio Massimo Zanzotto and Lorenzo Ferrone. 2017. Have you lost the thread? discovering ongoing conversations in scattered dialog blocks. *ACM Transactions on Interactive Intelligent Systems (TiiS)*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

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 1810.12885*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019a. BERTScore: Evaluating text generation with bert. In *International Conference on Learning Representations.*

Yuan Zhang, Jason Baldridge, and Luheng He. 2019b. PAWS: Paraphrase adversaries from word scrambling. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).*

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Association for Computational Linguistics.

Zheng Zhao, Shay B Cohen, and Bonnie Webber. 2020. Reducing quantity hallucinations in abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2237–2249.

Yanan Zheng, Jing Zhou, Yujie Qian, Ming Ding, Chonghua Liao, Li Jian, Ruslan Salakhutdinov, Jie Tang, Sebastian Ruder, and Zhilin Yang. 2022. FewNLU: Benchmarking state-of-the-art methods for few-shot natural language understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 501–516.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

Xiyou Zhou, Zhiyu Chen, Xiaoyong Jin, and William Yang Wang. 2021. Hulk: An energy efficiency benchmark platform for responsible natural language processing. In *Proceedings of the*

16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 329–336.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.