GENIE Toward Reproducible and Standardized Human Evaluation for Text Generation

Daniel Khashabi[3*] **Gabriel Stanovsky**[1,4] **Jonathan Bragg**[1] **Nicholas Lourie**[5*] **Jungo Kasai**[2]
Yejin Choi[1,2] **Noah A. Smith**[1,2] **Daniel S. Weld**[1,2]
[1]Allen Institute for AI [2]University of Washington [3]Johns Hopkins University [4]Hebrew University of Jerusalem [5]New York University

Abstract

While often assumed a gold standard, effective human evaluation of text generation remains an important, open area for research. We revisit this problem with a focus on producing consistent evaluations that are *reproducible*—over time and across different populations. We study this goal in different stages of the human evaluation pipeline. In particular, we consider design choices for the annotation interface used to elicit human judgments and their impact on reproducibility. Furthermore, we develop an automated mechanism for maintaining annotator quality via a probabilistic model that detects and excludes noisy annotators. Putting these lessons together, we introduce GENIE: a system for running *standardized* human evaluations across different generation tasks. We instantiate GENIE with datasets representing four core challenges in text generation: machine translation, summarization, commonsense reasoning, and machine comprehension. For each task, GENIE offers a leaderboard that automatically crowdsources annotations for submissions, evaluating them along axes such as correctness, conciseness, and fluency. We have made the GENIE leaderboards publicly available, and have already ranked 50 submissions from 10 different research groups.[1] We hope GENIE encourages further progress toward effective, standardized evaluations for text generation.

1 Introduction

While the emergence of powerful language models (Radford et al., 2019; Raffel et al., 2020; Lewis et al., 2020) has made *text generation* omnipresent, effective evaluation of the resulting systems' performance on open-ended generation tasks remains a challenge. This has motivated adoption of human evaluation in recent works (Celikyilmaz et al., 2020; Fabbri et al., 2021), even though it poses several challenges (Clark et al., 2021; Karpinska et al., 2021). First, the estimates of system performance are not *reproducible*—over time and various annotator populations. Additionally, the setups are not *standardized*. Different works use different annotation interfaces, even those working on the same dataset, despite substantial efforts needed for building an appropriate annotation interface and guidelines to extract quality human annotations and filter out noisy annotators.

This work presents an investigation toward reliably repeatable and standardized human evaluation. First and foremost, we study the reproducibility of human annotations, in two stages of the annotation pipeline. We study this goal empirically as a func-

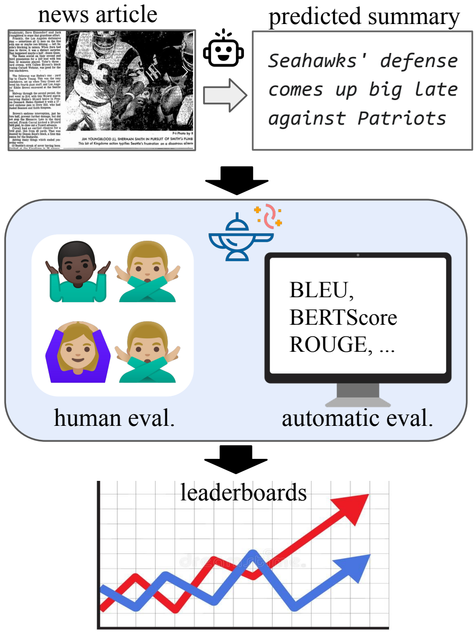


Figure 1: The GENIE architecture for evaluating text generation tasks, with a summarization example. Similar to automatic leaderboards, model developers submit their predictions (top). GENIE then evaluates with a standard human evaluation as well as with automatic metrics (center). These scores are then used to rank and track systems' performance across time (bottom).

*Work done at Allen Institute for AI.
[1]https://genie.apps.allenai.org

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11444–11458
December 7-11, 2022 ©2022 Association for Computational Linguistics

tion of various design choices (§4) such as the way the judgments are aggregated. We then propose a probabilistic framework for detecting malicious annotators (§5) and isolating their annotations from the resulting performance estimates.[2]

Guided by the earlier studies, we present GENIE (Figure 1)—a framework for human evaluation of text generation, which scales to a variety of tasks and datasets (§6). GENIE posts model predictions to a crowdsourcing platform,[3] where human annotators evaluate them according to predefined, dataset-specific guidelines. We describe mechanisms introduced into GENIE to quantify annotator variance and spread the annotations across various days, showing that GENIE achieves reliable scores on the studied tasks. To show its applicability, we instantiate GENIE with leaderboards for several popular text generation datasets in English from four diverse tasks—machine translation, question answering, summarization, and commonsense reasoning—and invite developers to extend it with more datasets. Since its deployment, GENIE has analyzed and ranked about 50 submissions from 10 different groups across all of our tasks, indicating the interest in standardized human evaluation.

The GENIE infrastructure opens the door for three avenues of research: (1) GENIE provides *developers* of text-generation models with the ease of the "leaderboard experience," alleviating the evaluation burden while ensuring high-quality, standardized comparison against previous models. (2) GENIE facilitates the study of *human evaluation interfaces* (Nenkova and Passonneau, 2004; Liu et al., 2016; Bragg et al., 2018; Shapira et al., 2019), addressing challenges such as annotator training, inter-annotator agreement, and reproducibility, all of which can be integrated into GENIE to compare against other evaluation metrics on past and future model submissions. (3) GENIE helps developers of *automatic evaluation metrics* (Zhang et al., 2020b), by serving as a hub of model submissions and associated human scores.

2 Related Work

We survey relevant work on automatic and human-in-loop evaluation of text generation. See Welty et al. (2019); van der Lee et al. (2019); Celikyilmaz et al. (2020) for further in-depth discussion.

(Semi-)automatic Metrics Many researchers have proposed automated metrics for text generation tasks, such as BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005). These metrics initially correlated well with human judgments for contemporary models (Papineni et al., 2002; Doddington, 2002; Coughlin, 2003), though the correspondence breaks down as they become targets for optimization (Callison-Burch et al., 2006; Sun et al., 2019) or as models become increasingly powerful (Ma et al., 2019; Edunov et al., 2020). Several more recent approaches aim to learn automated metrics for text generation tasks, including for image description (Vedantam et al., 2015), paraphrasing (Sellam et al., 2020), and abstractive question answering (Chen et al., 2020). Such progress in automatic metrics is incorporated into recent leaderboards (Kasai et al., 2021). We integrate some of these metrics into our proposed system to track their correlation with human evaluation.

Human Evaluation of Language Given the limitations of automatic metrics, much prior work has developed ways to conduct human evaluation of language generation in general, and machine translation in particular. Human evaluation for machine translation (Graham et al., 2013, 2014; Sakaguchi and Van Durme, 2018; Freitag et al., 2021) typically involves crowdsourcing where qualified crowd workers score output translations given the reference text. Results from manual evaluation are used as the primary metric in recent WMT competitions (Bojar et al., 2016, 2018; Barrault et al., 2020). However, to date, human evaluation efforts are typically conducted (1) on individual tasks such as machine translation, (2) by individual researchers with potentially varying design decisions, making results incomparable across evaluations, or (3) through shared tasks such as WMT, which force synchronization across teams for evaluation, slowing progress. As a result, most of the research on model development still evaluates models solely on automatic metrics such as BLEU (Papineni et al., 2002). GENIE relaxes these limitations by providing a continually-running leaderboard across language generation tasks with shared high-quality human evaluation templates.

Human-in-the-loop Evaluation There are a few recent and concurrent leaderboards that incorporate manual analysis, tending to focus on individual tasks. For example, HYPE (Zhou et al., 2019) is an evaluation platform for image generation, ChatE-

[2] Code implementing the model is available at `https://github.com/allenai/genie-worker-scoring`.

[3] We use Amazon Mechanical Turk.

val (Sedoc et al., 2019) is an evaluation platform for chatbots, and, more recently, Zellers et al. (2021) present a leaderboard for the advice generation task introduced in their work. DynaBench (Kiela et al., 2021) is a related multi-task leaderboard but uses changing, adversarially-created datasets that do not support our goal of controlled model comparison across time. HUME (Hashimoto et al., 2019) was proposed as an evaluation metric for summarization and dialog which combines human annotations with automatic metrics for diversity and quality. STORIUM (Akoury et al., 2020) was introduced for human-in-the-loop generation and evaluation of long open-ended stories as a computer game. Concurrently, Gehrmann et al. (2021) introduced GEM, a workshop for participant-driven evaluation of language generation tasks. While such workshops inspire progress toward common goals, synchronized evaluations, often only once per year, likely slow progress. We take the view that evaluations on a more frequent, rolling basis will give researchers more flexibility. To the best of our knowledge, GENIE is the first crowdsourced human-in-the-loop system that supports task leaderboards and is backed by principled design to ensure scoring reliability of human evaluations.

3 GENIE Principles for Human Evaluation of Generative Models

There are many ways to run human evaluations. Reflecting on what's needed to compare text generation models across time, we formulated the following principles to guide our design choices.

Application-Motivated Ultimately, the evaluation's purpose is to identify useful models and techniques. Thus, it should measure something informative about the their usefulness in applications (such as a generated text's correctness or fluency).

Reproducible To compare different models over time, the evaluation must be reproducible. If repeated, it should give largely the same results. For example, results should hold across different groups of annotators, and remain stable across appropriate lengths of time.

Interpretable The evaluation should help a researcher understand how the system behaves, and thus must measure an aspect of the system that is easy to understand. An evaluation which ranks models but isn't interpretable has limited usefulness, since different applications might prioritize different things and researchers must navigate cost-benefit trade offs between more expensive, higher performing models and cheaper ones.

Scalar The evaluation should produce an absolute scalar measurement of the model performance (rather than a relative or comparative one) that facilitates comparison of a new model to all those previously evaluated.

Quantified Uncertainty All measurements are subject to uncertainty, including human evaluations. Thus, when comparing evaluations, we should consider how confident we can be that the resulting measurement is close to the true, latent measurement based on a more complete population of inputs and human annotators.

Rolling Given rapid recent advances in natural language generation, it is essential to develop easily-accessible evaluation platforms for frequent model evaluations that do not require competing teams to synchronize with each other.

Extensible Evaluation of NLP models is actively evolving, as new datasets are introduced and more is learned about how best to conduct human evaluation. Therefore, an evaluation framework should be easily extensible to new tasks or the latest practices.

Next, we empirically study design decisions along the aforementioned evaluation desiderata.

4 Design Decisions for Consistent Human Evaluations

When designing an evaluation, some questions can be answered with principles, while others must be answered empirically. We investigate several questions around the prompt design that commonly occur across various tasks and impact evaluations' reproducibility and confidence.

(Q_1) *Granularity of the elicitation:* We examine two kinds of labels: (a) binary, and (b) Likert for 5 categories: Strongly agree, agree, neutral, disagree, and strongly disagree.

(Q_2) *Aggregation of per-example labels:* Given multiple labels per example, we investigate aggregating by (a) averaging their scores, and (b) taking a majority vote.

(Q_3) *Labels per example:* for a fixed annotation budget, we compare collecting (a) 3 labels per annotation example (multilabeling), with (b)

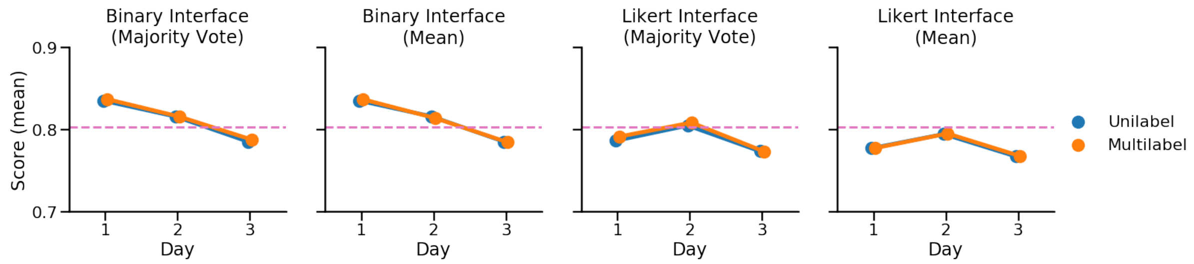


Figure 2: Label aggregation variants across three different days for the GENIE ARC-DA leaderboard. Using Likert scales yield lower inter-day variation. The horizontal red dashed line (0.803) denotes the annotations by an expert annotator intimately familiar with the task.

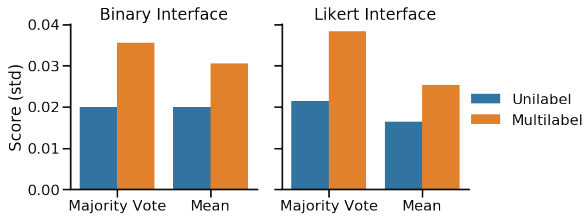


Figure 3: Standard deviation (STD) of different labeling strategies. *Unilabeling* yields lower variance and hence, better stability across different populations of annotators (on different days).

one label for three times[4] as many annotation examples (unilabeling).

Case Study: Comparing Evaluation Designs for Open-domain Question Answering (ARC-DA To study these design choices, we evaluated T5-11B (Raffel et al., 2020) on the development set of ARC-DA (Clark et al., 2021), a generative question answering dataset (see §7.1 further details). We used modified versions of the same annotation interface as Bhakthavatsalam et al. (2021) (Figure 4). Each evaluation was run once with a Likert scale and once with a binary scale. All instances ($n = 360$) were annotated by 3 annotators, repeated three times across different weekdays.[5] Then the quality judgments were mapped to numerical values.[6] To produce the unilabeling and multilabeling results, we simulated these policies by randomly sampling with replacement for 500 rounds, either a random 1/3 of the total number of examples (multilabeling) or 1/3 of the total number of annotations for each example (unilabeling).[7]

Figure 2 compares the reproducibility of different setups across time. Each subplot represents a choice of (Q_1) scale (binary/Likert), and (Q_2) aggregation (mean/majority-vote). We compare these setups across subplots, and within subplots compare (Q_3) unilabeling and multilabeling. The choices of scale and aggregation appear to have little effect on the evaluation, with all combinations broadly stable across days, though Likert elicitation with mean aggregation is slightly more stable.

Figure 3 compares the variance for all possible combinations. The choices of scale and aggregation appear to have little effect, though the Likert scale with mean aggregation may have the lowest variance. The biggest impact comes from unilabeling, which noticeably reduces the variance in comparison to multilabeling across all scenarios. This observation is consistent with previous work demonstrating the effectiveness of unilabeling for model training (Lin et al., 2014), but deviates from how annotations are often done in NLP (van der Lee et al., 2019). Our finding suggests that unilabeling is a promising strategy for model evaluation.

Overall, *unilabeling* with *Likert scales* and *mean aggregation* appears most reliable among all configurations for ARC-DA, and therefore we use this configuration in GENIE. Moreover, for the main leaderboard evaluations we use 3–7 times more samples, and expect even less variation. Our analysis shows that these design choices provide a good starting point for reproducible experiments with confident estimates.

5 Monitoring Annotation Quality

Despite strict qualification requirements, in our early experiments some annotators chose arbitrary labels after initially choosing correct ones. While a small percentage, these annotators complete a disproportionate share of tasks and significantly impact evaluations. To solve this problem, we built a monitoring system with two components: automatically generated *test questions* and an unsupervised

[4]"3 times" is to ensure the same amount is annotated in both scenarios for fair comparison.

[5]For consistency, experiments were launched at 10am PST.

[6] The binary scale was mapped to 0 and 1, while the Likert scale was mapped to 0, 0.25, 0.50, 0.75, and 1.

[7]This budget ensured that the number of sampled examples was at most the total number of examples, for unilabeling.

scoring model.[8]

Test Questions Because noisy annotators could favor marking examples as correct or incorrect, test questions need both *positive* and *negative* examples. For positive examples, we replaced model predictions with gold responses. For negative examples, we cyclically permuted the gold generations, so no example was matched with its original. Thus, the negative examples look correct at a glance, but almost never are.

Scoring Model Manually reviewing annotations can be time consuming and error prone, so we automate this process with a *scoring model* to infer if workers have acceptable accuracy. Probabilistic models of annotation have been richly studied (Hovy et al., 2013; Passonneau and Carpenter, 2014; Paun et al., 2018). Much prior work uses worker agreement to identify noisy annotators. Since we use unilabeling (§4), workers annotate disjoint sets of examples and these methods are not applicable. Instead, we use a similar probabilistic model but applied to predict how often workers correctly answer the test questions. Such a model must be *unsupervised*, since new tasks won't have identified noisy annotators, *interpretable*, since parameters like confidence thresholds must be set a priori, and *sequential*, so noisy annotators can be detected as soon as there is enough evidence.

In our model, each worker, w, answers n_w test questions. The number of correctly answered test questions, X_w, is binomially distributed with mean P_w. Each P_w comes from a mixture of betas prior. Thus, noisy and non-noisy annotators can be modeled with different mixture components.

$$\begin{aligned} Z_w &\sim \text{Categorical}(\theta_1, \ldots, \theta_k) \\ P_w &\sim \text{Beta}(\alpha_{Z_w}, \beta_{Z_w}) \\ X_w &\sim \text{Binomial}(P_w, n_w) \end{aligned}$$

We compare two definitions of noisy annotators. The **rate** criterion defines them as workers with an accuracy (P_w) below a threshold (90%). The **class** criterion defines them as workers whose latent class (Z_w) corresponds to any mixture component besides the one with highest expected accuracy.

We fit the model parameters, $\theta_i, \alpha_i, \beta_i$, for mixture components $i = 1, \ldots k$, with maximum likelihood via the EM algorithm (Dempster et al., 1977) for mixture models (Murphy, 2012). Then, we infer a posterior distribution for each worker's accuracy (P_w) and latent class (Z_w) given the number of questions they answered correctly (X_w). Since the prior is a mixture of conjugate distributions, the posteriors have a closed-form (Diaconis and Ylvisaker, 1979).

To adapt the Likert responses for this model, we binarize them at 0.5. Positive and negative test questions are modeled independently, and annotators are considered noisy if they are noisy on either. Since the difficulty of annotating different tasks varies, each GENIE task is modeled separately. Finally, to stabilize the EM algorithm and resulting parameter estimates, we augment the worker responses with pseudo-data. See Appendix B.1 for the full technical details.

Detecting Noisy Annotators for WMT21 To test GENIE in a real-world scenario, we used it to evaluate the 24 systems submitted to WMT21 and several additional baselines on German-to-English translation (Akhbardeh et al., 2021). For the evaluations, the GENIE leaderboards used 5% of examples as positive and 5% as negative test questions. We manually reviewed test question statistics to identify and remove 5 noisy annotators from a pool of 88 (5.7%). As in our preliminary experiments, these noisy annotators represented a small fraction of annotators; however, we had previously found such annotators could annotate up to 50% of the HITs.[9] By identifying and removing them, we prevented such a negative impact on our WMT evaluations.

Simulation Study Even a fairly large real-world evaluation encounters only a few noisy annotators. So, we complement our WMT21 case study with simulations based on it, where we can run more trials and know the ground truth.

We split the WMT21 annotations chronologically into validation and test sets. The validation set was used during model development, while we evaluated the models by simulating 25 rounds of annotation based on the test set's statistics.[10] Similarly to the annotation models discussed in Karger et al. (2011), each worker was independently designated as noisy and then assigned a rate at which

[8]We also tried clustering approaches that don't require test questions; however, they did not have good performance.

[9]A HIT (Human Intelligence Task) represents a single, self-contained, virtual task that a crowd worker can work on and collect a reward for completing.

[10]The test set contained only 2 noisy annotators, too few to compute reliable metrics in a direct evaluation.

they labeled test questions correctly. Based on the validation data's statistics, for each round we drew the noisy annotator probability uniformly from 1–10%, and each annotator's probability of being correct uniformly between 0–50% for noisy annotators and 95–100% for the rest. The model predicted annotators as noisy if the posteriors for Z_w and P_w assigned them at least 99% probability of being noisy annotators under the **rate** or **class** criteria. We computed precision and recall across all the simulations, bucketing workers by how many test questions they answered.

Model	**Prior**	**k**	**Precision / Recall**		
			1-4	**5-14**	**15+**
class	fixed	2	100/ 24	100/ 91	100/ 87
	learned	2	100/ 15	100/ 77	100/100
rate	Jeffreys	1	68/ 85	93/ 93	100/ 98
	uniform	1	56/ 86	93/100	100/100
	fixed	1	100/ 32	100/ 94	98/100
	learned	1	100/ 14	100/100	96/100
	fixed	2	86/ 33	100/100	100/ 97
	learned	2	100/ 12	100/ 92	100/100

Table 1: Noisy annotator detection models' precision and recall for workers who answered different numbers of test questions (1-4, 5-14, and 15+). Precision and recall were averaged across multiple simulations.

Table 1 shows the simulation results. In addition to varying the number of components (**k**) in the *learned* priors, we also compared against uninformative priors (the *Jeffreys* and *uniform* priors), and informative priors (*fixed*). Noisy annotators lose the chance to answer additional test questions, thus it's critical that models have high *precision* when marking workers as noisy. The uninformative priors suffer from low precision, assigning too much probability to a worker being a noisy annotator. The informed and learned priors both perform well, with high precision and good recall—in some cases identifying almost all noisy annotators with fewer than 15 test questions. The learned priors have the additional advantage that they can adapt to different distributions by pooling information across annotators. Based on these results, the 2-component learned rate and class models have proven to be strong candidates for application.

6 Automatically Managing Human Evaluation Leaderboards

This section reviews the GENIE system, which automates much of the management of text generation leaderboards with human evaluations. While we note that some human management, such as providing support and handling disputes, should never be fully automated, GENIE alleviates much of the overall burden. The next section (§7) describes its instantiation into the GENIE leaderboards for four diverse text generation tasks.

At a high level, the GENIE system coordinates a leaderboard UI, data processing backend, and crowdsourcing campaigns on Amazon Mechanical Turk. After retrieving newly uploaded submissions, the backend computes automatic metrics. Upon success, the backend then creates annotation tasks on Amazon Mechanical Turk (AMT) using AMTI[11] (A Mechanical Turk Inferface), an open-source Python package for working with AMT.

Each leaderboard is a separate instance of the system, with its own crowdsourcing templates, including instructions, examples, and prompts (see §7). Following our observations in §4, all templates use Likert scales which are then mapped to real-valued scores (cf. footnote 6) and averaged.

The system also maintains a history of past annotations (per-instance and per-worker), updating statistics after each evaluation. This has several immediate and future benefits: worker statistics enable spam detection (§5), while the annotations can be used for future studies on human evaluation.

These components enable the following features:

Extensibility New tasks can be modularly added to the GENIE system, creating new leaderboards. Each task requires a crowdsourcing template and a code object specifying how to push model predictions into and pull workers' annotations from the crowdsourcing templates. We release an extensible open-source annotation template library,[12] seeded with the four task templates used in this work.

Uncertainty Quantification To better inform model comparisons, we report scores with uncertainty estimates. Bootstrap resampling (samples with replacement from the observed annotations) provides the 95% confidence intervals for the estimated submission quality scores, as commonly done in machine translation (Koehn, 2004).

Human Evaluations: Uncertainty vs Cost To balance confidence with affordability, the system evaluates a subsample of the test sets. This subset is random, but fixed to reduce the variance between

[11] https://github.com/allenai/amti

[12] https://github.com/allenai/evaluation-interfaces

model comparisons. Sentence-level tasks, such as translation of sentences, cost less to annotate per example. Depending on task difficulty, we adjust the pay rate per HIT such that we are paying workers at a higher rate than 15 USD per hour. For these tasks we annotate 800 instances at a cost of ∼\$600 per submission (standard error $< 1.77\%$). For larger tasks, we evaluate 300 instances costing ∼\$350 per submission (standard error $< 2.89\%$).[13] These evaluations are much larger than what was previously done, e.g., 100 instances for MT in Ma et al. (2018) or around 100 instances for summarization (Kim et al., 2019; Hardy et al., 2019; Kryscinski et al., 2019; Fabbri et al., 2021).

Automatic Metrics To supplement human evaluations, we compute recent and popular automatic metrics for each task: METEOR (Banerjee and Lavie, 2005), ROUGE (Lin et al., 2006), BLEU (Papineni et al., 2002), SacreBLEU (Post, 2018), BLEURT (Sellam et al., 2020) and BERTScore (Zhang et al., 2020b). Integrating these metrics into GENIE enables researchers to examine their correlation with human judgments as well as observing trends as more models are submitted.

Quality Control To ensure annotation quality, annotators must pass strict qualifications requirements[14] and task-specific qualification tests based on a subset of the questions derived from the task's training data. These tests check that the workers have carefully read the instructions and are comfortable with annotating instances of the particular task. In addition, we replace 5% of examples with positive and another 5% with negative test questions, which we analyze with the 2-component learned class model between submission evaluations, as described in §5. Accordingly, noisy annotations are excluded from results and annotators from the pool of eligible workers. Lastly, to eliminate variability from evaluating at different times (weekend vs. weekdays, different work hours), we publish the AMT tasks on weekdays at 10am Pacific Time.

[13] See Appendix C for a discussion of standard error.

[14] I.e., 5000 completed HITs, a 99% assignment approval rate, and being based in a country with a population predomninantly of native English speakers (e.g., USA, Canada, UK, Australia) since our initial set of tasks focuses on English.

Task	Dataset	Domain	Train	Dev	Test
Question Answering	ARC-DA	basic science	$1.4k$	$0.4k$	$1.5k$
Summarization	XSUM	News	$200k$	$11k$	$11k$
Commonsense	aNLG	ROCStories	$170k$	$1.5k$	$3k$
Machine Translation	WMT19 DE-EN	News	$38.7m$	$3k$	$3k$
Machine Translation	WMT21 DE-EN	News	$101m$	$3k$	$1k$

Table 2: Datasets currently available in GENIE, along with their domain and size by task type.

7 The GENIE Leaderboards

7.1 Tasks and Datasets

We integrate in GENIE datasets from four diverse text-generation tasks, representing longstanding challenges, as outlined below. We focus on English language datasets, mostly due to easy integration with crowdsourcing platforms. In the future, we hope to integrate other new datasets, particularly other languages. GENIE is easily extensible; it uses community datasets and metrics via the open-source Datasets library.[15] The templates for all tasks are exemplified in Figure 4.

Question Answering Given an input question about a given context, the system is expected to provide the answer in natural-language form. We use the ARC-DA dataset,[16] which contains questions about subjects from elementary-school science exams. See Figure 4 for an example.

Commonsense Reasoning Given an input scenario, the task is to generate a plausible explanation, according to typical real-world human behavior and understanding. We use αNLG (Bhagavatula et al., 2020), a dataset for the conditional generation task of explaining given observations in natural language. For evaluation, we use a template and instructions that are similar to those used by Bhagavatula et al. (2020), as shown in Figure 4b.

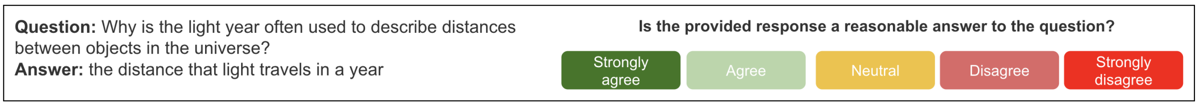
Machine Translation The task is to generate a translation in a target language given a text in a source language. Here we use the recent WMT19 and WMT21 datasets with publicly available system outputs (Barrault et al., 2019; Akhbardeh et al., 2021).[17] To ensure the generated text is evaluated by native speakers, we focus on German-to-English translation (DE-EN), and leave the expansion to

[15] Huggingface's Datasets repository

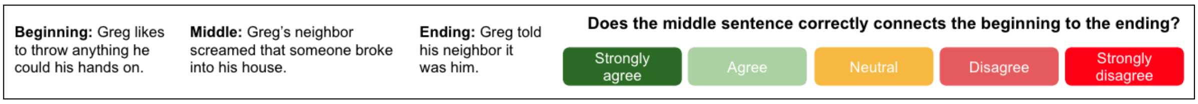
[16] ARC-DA dataset.

[17] WMT21 predictions repository.

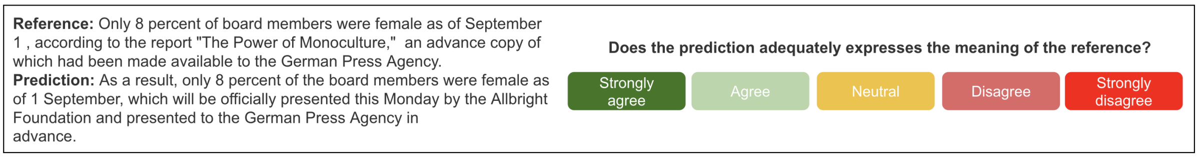
(a) Question Answering (ARC-DA).



(b) Commonsense Reasoning (αNLG), adapted from Bhagavatula et al. (2020).



(c) Machine Translation (WMT19 and WMT21 DE-EN), adapted from Barrault et al. (2019).



(d) Summarization (XSUM), adapted from Chaganty et al. (2018). Here, Summary A is the gold label while Summary B is model-predicted text. We permute this randomly between instances so that the annotators are blind to which one is gold.

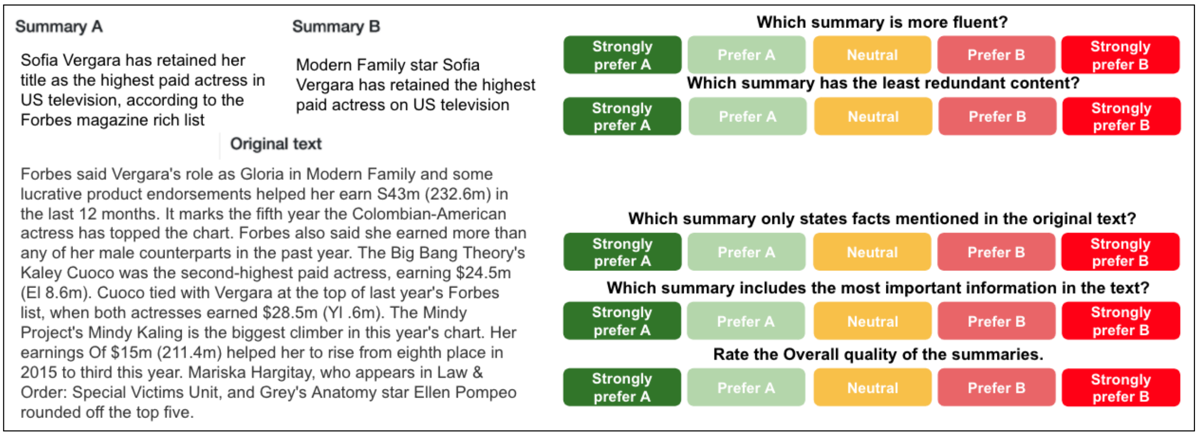


Figure 4: Annotation interfaces for the datasets of four tasks integrated in GENIE.

other language pairs as future work. Importantly, WMT19 and WMT21 DE-EN test data only contain text that was originally in German (Barrault et al., 2019), avoiding overestimating the quality of translation systems due to translationese effects (Toral et al., 2018; Graham et al., 2019; Edunov et al., 2020). We follow the WMT human evaluation template to assess sentence-level translation quality against the reference (Barrault et al., 2019). The one difference is that, consistent with the other GENIE tasks, we use a five-category Likert scale instead of a continuous one in WMT. See Figure 4c.

Summarization The model is expected to generate a summary of the key points mentioned in a given paragraph. Here we use XSUM (Narayan et al., 2018), a news summarization dataset. We chose XSUM over alternative datasets for text summarization (e.g., CNN/DM, Hermann et al., 2015) since the task involves more abstractive summaries and hence more difficult to evaluate with existing automatic metrics. For evaluating this task we use a template similar to that of Chaganty et al. (2018); Fabbri et al. (2021) and measure different aspects of quality (redundancy, fluency, conciseness, etc.) that have traditionally been of interest (McKeown and Radev, 1995). See Figure 4d for an example.

7.2 Evaluating GENIE Baselines

Here we evaluate several baseline models for each dataset using the GENIE evaluation pipeline.

Models We use models that are known to perform strongly for each of our tasks. For all tasks but machine translation, we train and evaluate T5 (11B; Raffel et al., 2020), a powerful text-generation model that has shown promising results on a wide variety of text generation tasks.

For WMT we evaluate other specialized models instead of T5, which is pre-trained only on English (Raffel et al., 2020). For WMT21 DE-EN, we evaluate all publicly available shared task sub-

ARC-DA (Question Answering)				
Systems	Human	ROUGE	SacreBLEU	BLEURT
UnifiedQA (ARC-DA/MC+IR)	**80.8**$^{+2.1}_{-2.2}$	**63.1**	**22.2**	**29.40**
UnifiedQA (ARC-DA+IR)	75.3$^{+2.3}_{-2.4}$	61.3	19.7	27.53
T5 (11B)	66.0$^{+2.6}_{-2.5}$	47.4	12.8	1.6
T5 (3B)	60.9$^{+2.9}_{-3.0}$	43.2	11.7	-5.2
WMT21 (Machine Translation)				
Systems	Human	ROUGE	SacreBLEU	BLEURT
Watermelon	**75.7**$^{+2.0}_{-2.0}$	**64.8**	**34.5**	**34.7**
VolcTrans-AT	**75.2**$^{+2.0}_{-2.0}$	**64.8**	34.4	34.6
HUMAN	**75.2**$^{+2.0}_{-2.0}$	59.3	29.5	30.0
GENIE-large-6-6	70.4$^{+1.9}_{-2.0}$	63.3	32.4	31.3
GENIE-base-6-6	69.0$^{+2.1}_{-2.1}$	63.3	31.8	28.2
GENIE-base-3-3	65.3$^{+2.3}_{-2.3}$	62.7	31.2	23.9
GENIE-base-1-1	50.7$^{+2.3}_{-2.4}$	59.3	27.0	-0.2
αNLG (Commonsense Reasoning)				
Systems	Human	ROUGE	SacreBLEU	BLEURT
T5 (11B)	**75.9**$^{+1.1}_{-1.0}$	**44.6**	**19.5**	**-22.2**
GPT-2 (unsupervised)	45.1$^{+1.2}_{-1.3}$	19.7	1.8	-84.5
XSUM (Summarization)				
Systems	Human overall	ROUGE	SacreBLEU	BLEURT
Anonymous (ARR submission)	**50.6**$^{+3}_{-3}$	35.8	13.9	-21.7
Pegasus	**48.7**$^{+3.1}_{-3.4}$	**39.1**	16.7	-17
T5 (11B)	**47.5**$^{+3.3}_{-3.3}$	37.9	**17.1**	**-14.3**

Table 3: Summary of evaluating several existing, strong models on each dataset with GENIE. The highest numbers and their confidence interval (CI) in each column are indicated in **bold**. The scores given by crowd workers are indicated with blue color. We evaluated all participating systems from WMT21 but only show the top 3 systems as well as our GENIE transformer baselines for clarity. See Table 6 (appendix) for more metrics and WMT19 results.

missions (see footnote 17). Additionally, we train and evaluate four transformer-based baselines with varying sizes: GENIE-large-6-6 (transformer large with a 6-layer encoder and a 6-layer decoder), GENIE-base-6-6, GENIE-base-3-3, and GENIE-base-1-1.[18] These models are trained solely on the given training data without ensembling, backtranslation, or any other data augmentation method, to support future research in low-compute settings.

In addition to the above baselines we evaluate specialized baselines for each task. For αNLG, we evaluate an unsupervised baseline (Qin et al., 2020) based on GPT-2 (Radford et al., 2019). For summarization, we evaluate Pegasus (Zhang et al., 2020a). For ARC-DA, we evaluate a fine-tuned version of UnifiedQA (Khashabi et al., 2020).

Results The results are summarized in Table 3. The human judgment scores for each task are calculated with our described pipeline (§6). Even though we have evaluated strong baselines for each task, the machine responses are far from what human judges consider perfect. In the WMT21 task, the transformer baselines are ranked in the expected order: large-6-6, base-6-6, base-3-3, followed by base-1-1. These results support the validity of our evaluations. We defer any further study of the correlations between human judgments and automatic metrics for future work since such a study would require more models to be evaluated.

8 Limitations

As with other works which deal with human annotation, the results generated via our evaluation framework will have inherent variability. While we tried to mitigate sources of variation in various ways (see §5,6), some are bound to remain and are hard to account for. These include, for example, selection bias in the pool of annotators that choose to work on our tasks, who may come from specific countries and social status and select for certain tasks and their templates. We welcome future evolution of all parts of the GENIE architecture, including its evaluation metrics.

9 Conclusion and Future Work

We introduce GENIE, a unified approach to human-in-the-loop evaluation of text generation over a wide set of text generation tasks. GENIE is open for use and will be adapted based on future adoption. We encourage submissions from all researchers interested in text generation models.

Acknowledgments

The authors would like to thank the leaderboard team at Allen Institute for AI, particularly Michal Guerquin and Sam Skjonsberg. We thank Peter Clark, Oyvind Tafjord and Daniel Deutsch for valuable feedback throughout this project. We are grateful to the many AMT workers whose contributions make human evaluation possible, and to the anonymous reviewers for their helpful feedback on this manuscript. This work was supported in part by DARPA MCS program through NIWC Pacific (N66001-19-2-4031) and research grant 2336 from the Israeli Ministry of Science and Technology.

[18] https://github.com/jungokasai/GENIE_wmt2021-de-en

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proc. of WMT*.

Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation. In *Proc. of EMNLP*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proc. of Intrinsic and Extrinsic Evaluation measures for Machine Translation and/or Summarization*.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proc. of WMT*.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proc. of WMT*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *Proc. of ICLR*.

Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. 2021. Think you have Solved Direct-Answer Question Answering? Try ARC-DA, the Direct-Answer AI2 Reasoning Challenge.

Rajendra Bhatia and Chandler Davis. 2000. A better bound on the variance. *The American Mathematical Monthly*, 107.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proc. of WMT*.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proc. of WMT*.

Jonathan Bragg, Mausam, and Daniel S. Weld. 2018. Sprout: Crowd-powered task design for crowdsourcing. *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *Proc. of EACL*.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey.

Arun Tejasvi Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalaution. In *Proc. of ACL*.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2020. MOCHA: A dataset for training and evaluating generative reading comprehension metrics. In *Proc. of EMNLP*, pages 6521–6532.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A Smith. 2021. All that's 'human'is not gold: Evaluating human evaluation of generated text. In *Proc. of ACL*.

Deborah Coughlin. 2003. Correlating automated and human assessments of machine translation quality. In *Proc. of MT Summit IX*.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Persi Diaconis and Donald Ylvisaker. 1979. Conjugate priors for exponential families. *The Annals of statistics*, pages 269–281.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proc. of HLT*.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2020. On the evaluation of machine translation systems trained with back-translation. In *Proc. of ACL*.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *TACL*, 9:391–409.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *TACL*, 9:1460–1474.

Sebastian Gehrmann, Tosin P. Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh D. Dhole, Wanyu Du, Esin Durmus, Ondrej Dusek, Chris Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Rubungo Andre Niyongabo, Salomey Osei, Ankur P. Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proc. of the Linguistic Annotation Workshop and Interoperability with Discourse*.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2014. Is machine translation getting better over time? In *Proc. of EACL*.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2019. Translationese in machine translation evaluation.

Hardy Hardy, Shashi Narayan, and Andreas Vlachos. 2019. HighRES: Highlight-based reference-less evaluation of summarization. In *Proc. of ACL*.

Tatsunori Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. In *Proc. of NAACL*, pages 1689–1701.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proc. of NeurIPS*.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with mace. In *Proc of NAACL*, pages 1120–1130.

David R. Karger, Sewoong Oh, and Devavrat Shah. 2011. Budget-optimal task allocation for reliable crowdsourcing systems. *CoRR*, abs/1110.3564.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The perils of using mechanical turk to evaluate open-ended text generation. In *Proc. of EMNLP*.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Lavinia Dunagan, Jacob Morrison, Alexander R. Fabbri, Yejin Choi, and Noah A. Smith. 2021. Bidimensional leaderboards: Generate and evaluate language hand in hand.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UnifiedQA: Crossing format boundaries with a single QA system. In *Proc. of EMNLP - Findings*.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proc. of NAACL*, pages 4110–4124.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of Reddit posts with multi-level memory networks. In *Proc. of NAACL*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proc. of EMNLP*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *Proc. of ACL*.

Chin-Yew Lin, Guihong Cao, Jianfeng Gao, and Jian-Yun Nie. 2006. An information-theoretic approach to automatic evaluation of summaries. In *Proc. of NAACL*.

Christopher H Lin, M Mausam, and Daniel S Weld. 2014. To re (label), or not to re (label). In *HCOMP*.

Angli Liu, Stephen Soderland, Jonathan Bragg, Christopher H. Lin, Xiao Ling, and Daniel S. Weld. 2016. Effective crowd annotation for relation extraction. In *Proc. of NAACL*.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance. In *Proc. of WMT Shared Task*.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proc. of WMT*.

Kathleen McKeown and Dragomir R Radev. 1995. Generating summaries of multiple news articles. In *Proc. of SIGIR*.

Thomas P. Minka. 2000. Estimating a dirichlet distribution. Technical report, MIT.

Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proc. of EMNLP*.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proc. of NAACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

Rebecca J Passonneau and Bob Carpenter. 2014. The benefits of a model of annotation. *TACL*, 2:311–326.

Silviu Paun, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. 2018. Comparing bayesian models of annotation. *TACL*, 6:571–585.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proc. of WMT*.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. In *Proc. of EMNLP*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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. *JMLR*.

Keisuke Sakaguchi and Benjamin Van Durme. 2018. Efficient online scalar annotation with bounded support. In *Proc. of ACL*.

Joao Sedoc, Daphne Ippolito, Arun Kirubarajan, Jai Thirani, Lyle Ungar, and Chris Callison-Burch. 2019. ChatEval: A tool for chatbot evaluation. In *Proc. of NAACL Demonstrations*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proc. of ACL*.

Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2019. Crowdsourcing lightweight pyramids for manual summary evaluation. In *Proc. of NAACL*.

Simeng Sun, Ori Shapira, Ido Dagan, and Ani Nenkova. 2019. How to compare summarizers without target length? pitfalls, solutions and re-examination of the neural summarization literature. In *Proc. of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 21–29.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the unattainable? reassessing claims of human parity in neural machine translation. In *Proc. of WMT*.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proc. of INLG*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *Proc. of CVPR*.

Chris Welty, Praveen Paritosh, and Lora Aroyo. 2019. Metrology for AI: From benchmarks to instruments.

Rowan Zellers, Ari Holtzman, Elizabeth Clark, Lianhui Qin, Ali Farhadi, and Yejin Choi. 2021. Turingadvice: A generative and dynamic evaluation of language use. In *Proc. of NAACL*, pages 4856–4880.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proc. of ICML*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020b. BERTScore: Evaluating text generation with bert. In *Proc. of ICLR*.

Sharon Zhou, Mitchell Gordon, Ranjay Krishna, Austin Narcomey, Li F Fei-Fei, and Michael Bernstein. 2019. HYPE: A benchmark for human eye perceptual evaluation of generative models. In *Proc. of NeurIPS*.

A Details on Model Engineering

Here we summarize the experimental details for building the models used in §7.2.

T5 models. For various datasets (except WMT which requires a multi-lingual model) we trained T5 models of different sizes: 11 billion parameters (11B) and 3 billion parameters (3B). We used the default hyperparameters on these frameworks: token-limits of size 512 and 100 for inputs and outputs sequences, respectively; learning rates of 1e-3 and batch sizes of 8. The models were trained for $100k$ steps on v3-8 TPUs which took about 24 hours to finish, on average. The checkpoint with the highest score on the dev set of each task was selected for evaluation.

GENIE WMT models. Tables 4 and 5 list hyperparameters for our GENIE transformer baselines. BPE with 32K operations is applied jointly to German and English text. All embeddings are shared.

Hyperparameter	Value
label smoothing	0.1
# max tokens	1024
dropout rate	0.1
encoder embedding dim	512
encoder ffn dim	2048
# encoder attn heads	8
decoder embedding dim	512
decoder ffn dim	2048
# decoder attn heads	8
max source positions	1024
max target positions	1024
Adam lrate	5×10^{-4}
Adam β_1	0.9
Adam β_2	0.98
lr-scheduler	inverse square
warm-up lr	1×10^{-7}
# warmup updates	4000
max epoch	7
# GPUs	8
length penalty	0.6

Table 4: Transformer-base `fairseq` hyperparameters and setting.

Hyperparameter	Value
label smoothing	0.1
# max tokens	4096
dropout rate	0.1
encoder embedding dim	1024
encoder ffn dim	4096
# encoder attn heads	16
decoder embedding dim	1024
decoder ffn dim	4096
# decoder attn heads	16
max source positions	1024
max target positions	1024
Adam lrate	5×10^{-4}
Adam β_1	0.9
Adam β_2	0.98
lr-scheduler	inverse square
warm-up lr	1×10^{-7}
# warmup updates	4000
max epoch	7
# GPUs	8
length penalty	0.6

Table 5: Transformer-large `fairseq` hyperprameters.

B Monitoring Annotation Quality

This appendix provides details on model construction and evaluation for §5.

B.1 Modeling

All *rate* models used $P_w = 0.9$ as the threshold for defining a noisy annotator. All *class* models used the mixture component with highest average accuracy to define non-noisy annotators. Workers were marked as noisy if the model assigned more than 99% probability to them being so. For reproducibility, we set Python and NumPy's random seeds to 0 at the beginning of our experiments.

Uninformative Priors Both uninformative priors had one beta mixture component. The *Jeffreys* model used a Jeffreys prior, or $\text{Beta}(\frac{1}{2}, \frac{1}{2})$. The *uniform* model used a uniform prior, or $\text{Beta}(1, 1)$.

Informative Priors The 1-component fixed rate model had one beta mixture component with parameters $\alpha = 4$ and $\beta = 1$. Both the 2-component fixed rate model and the 2-component fixed class model had two mixture components with probabilities 0.05 and 0.95 and parameters $\alpha = 0.5$, $\beta = 4.5$ and $\alpha = 9.5$, $\beta = 0.5$.

Learned Priors The learned prior models were all fit via the EM algorithm, as described below, and we tried 1 and 2 components for the rate model and 2 components for the class model.

Optimization To stabilize the EM algorithm and regularize the parameter estimates, we augmented with pseudo-data. To the data, we added 40 pseudo-workers, each completing 20 tasks: 36 annotators with 19 successes, and four noisy annotators with 1, 1, 5, and 10 successes, respectively. The EM algorithm was run with 10 initializations, each with up to 1,000 iterations and relative tolerance of $1e{-}6$ for stopping. Components were initialized with equal mixture probabilities, uniformly random means from 0 to 1, and concentration parameters drawn from a gamma distribution with a shape parameter of 2. Beta mixture components were fit using the Dirichlet-multinomial fixed point iterator from Minka (2000), 10,000 iterations and a relative tolerance of $1e{-7}$.

B.2 Evaluation

For evaluation, we simulated 25 rounds of annotation. In each round, the number of test questions were the counts from the test set of annotations, a fixed noisy annotator rate was drawn uniformly from 1% to 10%, a mean and concentration parameter for the beta distribution of noisy annotator's success probabilities was respectively drawn uniformly from 0 to 0.5 and 5 to 50, and a mean and concentration parameter for the beta distribution of regular annotator's success probabilities was respectively drawn uniformly from 0.95 to 1 and 100 to $1,000$. Each worker was assigned a noisy or regular annotator label and accordingly a success probability, then successes and failures were binomial distributed.

C Standard Error

Standard error quantifies the variability of an estimate, $\hat{\theta}$. Mathematically, the standard error is the estimate's standard deviation (as opposed to the standard deviation of a single sample). Often, the estimate is an average of multiple, independent samples, in which case the standard error is:

$$\frac{\sigma}{\sqrt{n}}$$

where n is the number of samples and σ is the standard deviation of a single sample. When the estimate is an average, it's approximately normally distributed due to the central limit theorem, making $\hat{\theta} \pm 1.96\frac{\sigma}{\sqrt{n}}$ an approximate 95% confidence interval.

The Bhatia-Davis inequality (Bhatia and Davis, 2000) bounds the variance of a random variable in terms of its upper bound, M, lower bound, m, and expectation, μ:

$$\sigma^2 \leq (M - \mu)(\mu - m)$$

Since the scores from our annotators are bounded between 0 and 1, the maximum standard deviation for any of them is 0.5. Moreover, if a model's score is 0.8 on average, then the maximum standard deviation for its annotations is $\sqrt{(1 - 0.8)(0.8 - 0)} = 0.4$. Dividing by $\sqrt{n}$ translates these into bounds on the worst-case standard error of our estimates:

$$\text{StandardError} \leq \sqrt{\frac{\mu(1 - \mu)}{n}},$$

where μ is the expected score.

ARC-DA (Question Answering)

Systems	Human	BERTScore	ROUGE	METEOR	SacreBLEU	BLEURT
T5 (11B)	$\mathbf{66.0}^{+2.6}_{-2.5}$	**92.4**	**47.4**	**33.1**	**12.8**	**1.6**
T5 (3B)	$60.9^{+2.9}_{-3.0}$	91.9	43.2	30.3	11.7	-5.2

WMT19 (Machine Translation)

Systems	Human	BERTScore	ROUGE	METEOR	SacreBLEU	BLEURT
FAIR	$\mathbf{69.8}^{+2.2}_{-2.2}$	95.3	66.0	**63.4**	**40.8**	**32.2**
GENIE-large-6-6	$\mathbf{70.6}^{+2.1}_{-2.1}$	**95.1**	**66.3**	63.1	40.7	26.3
GENIE-base-6-6	$65.0^{+2.3}_{-2.3}$	94.7	64.9	61.3	38.6	16.8
JHU	$66.0^{+2.2}_{-2.2}$	95.0	64.5	61.5	38.1	25.7

WMT21 (Machine Translation)

Systems	Human	BERTScore	ROUGE	METEOR	SacreBLEU	BLEURT
Watermelon	$\mathbf{75.7}^{+2.0}_{-2.0}$	**95.0**	**64.8**	**59.3**	**34.5**	**34.7**
VolcTrans-AT	$\mathbf{75.2}^{+2.0}_{-2.0}$	**95.0**	**64.8**	**59.3**	34.4	34.6
HUMAN	$\mathbf{75.2}^{+2.0}_{-2.0}$	94.8	59.3	54.3	29.5	30.0
GENIE-large-6-6	$70.4^{+1.9}_{-2.0}$	94.9	63.3	57.0	32.4	31.3
GENIE-base-6-6	$69.0^{+2.1}_{-2.1}$	94.7	63.3	56.8	31.8	28.2
GENIE-base-3-3	$65.3^{+2.3}_{-2.3}$	94.5	62.7	56.3	31.2	23.9
GENIE-base-1-1	$50.7^{+2.3}_{-2.4}$	93.3	59.3	50.9	27.0	-0.2

αNLG (Commonsense reasoning)

Systems	Human	BERTScore	ROUGE	METEOR	SacreBLEU	BLEURT
T5 (11B)	$\mathbf{75.9}^{+1.1}_{-1.0}$	**92.9**	**44.6**	**35.2**	**19.5**	**-22.2**
GPT-2 (unsupervised)	$45.1^{+1.2}_{-1.3}$	88.5	19.7	18.8	1.8	-84.5

XSUM (Summarization)

Systems	Human overall	Human conciseness	Human fluency	Human no-hallucination	Human informativeness	BERTScore	ROUGE	METEOR	SacreBLEU	BLEURT
Pegasus	$\mathbf{48.7}^{+3.1}_{-3.4}$	$\mathbf{52.0}^{+2.3}_{-2.5}$	$49.1^{+2.5}_{-2.4}$	$\mathbf{49.3}^{+2.9}_{-2.9}$	$\mathbf{49.2}^{+3.0}_{-2.8}$	91.9	39.1	35.4	16.7	-17
T5 (11B)	$\mathbf{47.5}^{+3.3}_{-3.3}$	$49.3^{+2.1}_{-1.9}$	$\mathbf{49.9}^{+2.7}_{-2.8}$	$\mathbf{49.4}^{+2.8}_{-2.8}$	$47.6^{+3.0}_{-2.8}$	**92.0**	37.9	**36.9**	17.1	**-14.3**

Table 6: Summary of evaluating several existing models on each dataset with GENIE. The highest numbers (and their CI) in each column are indicated in **bold**. The scores given by crowd workers are indicated with blue color. We evaluated all 24 systems from WMT21 but only show the top 3 systems as well as our GENIE transformer baselines here.