

Models in the Loop: Aiding Crowdworkers with Generative Annotation Assistants

Max Bartolo^{†‡} Tristan Thrush[‡] Sebastian Riedel^{†‡}

Pontus Stenetorp[†] Robin Jia^{§‡} Douwe Kiela[‡]

[†]UCL [§]USC [‡]Facebook AI Research

m.bartolo@cs.ucl.ac.uk

Abstract

In Dynamic Adversarial Data Collection (DADC), human annotators are tasked with finding examples that models struggle to predict correctly. Models trained on DADC-collected training data have been shown to be more robust in adversarial and out-of-domain settings, and are considerably harder for humans to fool. However, DADC is more time-consuming than traditional data collection and thus more costly per annotated example. In this work, we examine whether we can maintain the advantages of DADC, without incurring the additional cost. To that end, we introduce Generative Annotation Assistants (GAAs), generator-in-the-loop models that provide real-time suggestions that annotators can either approve, modify, or reject entirely. We collect training datasets in twenty experimental settings and perform a detailed analysis of this approach for the task of extractive question answering (QA) for both standard and adversarial data collection. We demonstrate that GAAs provide significant efficiency benefits with over a 30% annotation speed-up, while leading to over a 5x improvement in model fooling rates. In addition, we find that using GAA-assisted training data leads to higher downstream model performance on a variety of question answering tasks over adversarial data collection.

1 Introduction

Natural language processing has become increasingly reliant on large datasets obtained using crowdsourcing. However, crowdsourcing as an unconstrained annotation approach is known to result in machine-exploitable annotator artefacts (Jia and Liang, 2017; Schwartz et al., 2017; Gururangan et al., 2018; Geva et al., 2019), leading to poor out-of-distribution generalisation (Chen et al., 2016; Weissenborn et al., 2017; Yogatama et al., 2019; McCoy et al., 2019). Dynamic Adversarial Data Collection (DADC) aims to address these issues

After a slow start to the 2008–09 season, the Bruins won 17 of their next 20 games, leading many to see them as a revival of the "Big Bad Bruins" from the 1970s and 1980s. During the 2009 All-Star Weekend's Skills Competition, captain **Zdeno Chara** fired the NHL's then-fastest measured "hardest shot" ever, with a clocked in speed of **105.4 mph** (169.7 km/h) velocity. (Chara has since broken his own record three times, two of those on the same night.) The number of injured players in the season...



A: 105.4 mph

Q: What was the fastest shot ever?



Q: What was the fastest shot ever in 2009?

A: Zdeno Chara



Figure 1: Example interaction between an annotator and the models in the loop. The annotator selects an answer from the passage, for which the Generative Annotation Assistant (GAA) prompts a question. The annotator can then freely modify the question and/or answer, or generate another prompt. In the adversarial data collection setting, a model-in-the-loop provides predictions with the aim of encouraging annotators to find model-fooling examples. In the answer prompting setting, an answer suggestion is prompted by the assistive model instead of being selected by the annotator.

by introducing state-of-the-art models into the data collection loop and asking human annotators to produce examples that these models find challenging (Kiela et al., 2021). The intuition behind this approach is that it leads human annotators to better explore the space of possible examples. Previous work has found that DADC leads to improved model robustness on adversarial datasets (Nie et al., 2020; Bartolo et al., 2020), increased sample diversity (Bartolo et al., 2020; Wallace et al., 2021), better training data (Wallace et al., 2021) and better domain generalisation (Bartolo et al., 2021).

Despite these advantages, a downside to DADC is that it increases the human effort necessary to annotate a single example and thus the overall annotation cost. In fact, to date, only a limited number of

large-scale training datasets have been produced using DADC and its application has been primarily restricted to producing challenge sets or as additional training data to improve the performance of models already trained on non-DADC curated datasets. To make better use of DADC data, [Bartolo et al. \(2021\)](#) propose generating synthetic adversarial training sets to further improve model robustness. However, this approach inevitably limits example diversity as it relies on examples ultimately generated by a model with no additional human input, and provides no guarantees that useful synthetic examples would transfer across target adversary models of varying capabilities or across annotation rounds.

In this work, we propose assisting annotators by having generative models aid human annotators in the data collection loop. Concretely, we utilise a Generative Annotation Assistant (GAA) model that provides prompt suggestions to crowdworkers, while allowing full flexibility for edits and rewrites to support example generation while still allowing for human creativity as shown in [Figure 1](#). We explore GAAs in a broad range of experimental settings, including standard and adversarial data collection approaches, training on various source datasets, and employing sampling methodologies based on likelihood, adversarial feedback, and uncertainty. We showcase the value of this approach on the task of extractive question answering (QA), and find that GAAs can help improve both the standard and adversarial data collection paradigms. We find considerable efficiency gains, with over 30% observed annotation speed-ups, as well as improved data effectiveness with up to a 6.1F₁ improvement in downstream performance over adversarial data collection.

2 Related Work

2.1 Dynamic Adversarial Data Collection (DADC)

There is a rich body of recent work showing the benefits of dynamic adversarial data collection in model evaluation ([Yang et al., 2017](#); [Dua et al., 2019](#); [Dinan et al., 2019](#); [Nie et al., 2020](#); [Bartolo et al., 2020](#); [Kiela et al., 2021](#); [Wallace et al., 2021](#)), although the approach has been challenged for not necessarily leading to better generalisation on non-adversarial test sets ([Kaushik et al., 2021](#)) and being sensitive to the choice of model that was used in the loop ([Bowman and Dahl, 2021](#); [Phang et al., 2021](#)). This work builds on previ-

ous work in adversarial data collection methods for QA ([Bartolo et al., 2020](#)), and work investigating the use of question generation models to create synthetic adversarial data to improve QA model robustness ([Bartolo et al., 2021](#)).

2.2 Generative Model Annotation Support

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

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

2.3 Active Learning and Weak Supervision

Active learning approaches have been used to accelerate annotation ([Tsuruoka et al., 2008](#)), although this typically assumes access to a pool or stream of unlabelled data for which the learning algorithm can query labels ([Settles, 2009](#)). In our setting, no unlabelled questions are provided, necessitating the use of a generative model to suggest questions instead. Moreover, our annotators are free to edit

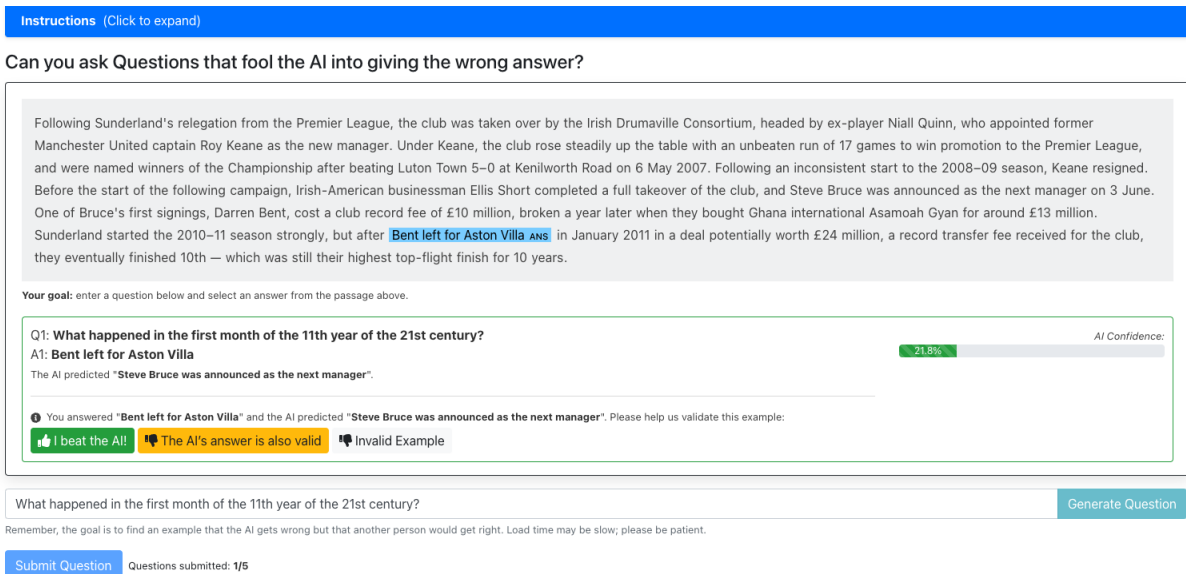


Figure 2: The Annotation Interface used for data collection. This example shows a question generated using a generative assistant trained on the AdversarialQA data and selected an adversarial sampler, which successfully allowed the annotator to beat the QA model in the loop.

and browse generated questions, whereas annotators in active learning typically only provide labels and have no choice in what to label. Some of our sampling and filtering strategies based on entropy are inspired by uncertainty sampling, a standard active learning algorithm (Lewis and Gale, 1994).

3 Experimental Setup

Our study focuses on the effects of incorporating generative annotation assistants, and understanding their interactions with annotators and discriminative models-in-the-loop in a DADC context for QA. We provide crowdworkers with a short passage from Wikipedia and ask them to write five questions and highlight the span in the passage that best answers the question for each (see Figure 2). We pay workers equally across experiment modes to avoid creating an incentive imbalance and pay out an additional bonus for each question that successfully beats the discriminative QA model i.e., for each question that the model fails to answer correctly. Finally, we validate all collected examples using a separate worker pool that also undergoes rigorous onboarding and validation. We ask three of these additional workers to report on the validity of each annotated example.

Selected Passages We select passages from KILT (Petroni et al., 2021) to allow for the possibility of future investigation into cross-domain and task transfer. We restrict KILT passages to

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

Model-in-the-Loop The discriminative QA model in the loop is ELECTRA_{Large} (Clark et al., 2020) trained on SQuAD1.1 and AdversarialQA, and enhanced using SynQA to improve adversarial robustness as investigated by Bartolo et al. (2021).¹ This model represents the best-performing model on the Dynabench (Kiela et al., 2021) leaderboard at the time of conducting this study, obtaining a word-overlap F₁ score of 94.5% on the SQuAD1.1 dev set, and represents the state-of-the-art on AdversarialQA achieving 77.6% on the $\mathcal{D}_{\text{BiDAF}}$ subset, 71.5% on $\mathcal{D}_{\text{BERT}}$, and 63.2% on $\mathcal{D}_{\text{RoBERTa}}$.

Generator-in-the-Loop For our generative models, we use the *fairseq* (Ott et al., 2019) implemen-

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

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

Question Sampling We investigate three different selection strategies for presenting the generated questions as prompts to annotators: i) *generator likelihood* samples candidates in the order prescribed by the generative model’s associated likelihood values; ii) *adversarial sampling* selects generated questions in order of the least word-overlap F₁ scores when queried against the discriminative QA model; and iii) *uncertainty sampling* is inspired by active learning and selects generated questions in order of the least span selection confidence when queried against the QA model. The latter two provide an interesting trade-off for exploration as we would expect the quality of the generated questions to be worse than if sampled based on likelihood. However, we hope that such prompts could serve to inspire annotators and provide a “starting point” beyond the answering capabilities of the QA model, irrespective of correctness. We hypothesise that modifying such examples might be a more effective process for annotators to undertake than when starting from higher quality but less model-confusing prompts, and investigate this question thoroughly.

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

corresponding generated questions.

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

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

Crowdsourcing Protocol We use Amazon Mechanical Turk to recruit workers for this study and run all experiments using Mephisto.² To ensure proficiency in English, crowdworkers are required to be based in Canada, the UK, or the US. They are also required to have a Human Intelligence Task (HIT) Approval Rate greater than 98%, have previously completed at least 1,000 HITs, and undergo a dedicated onboarding process. Workers were randomly assigned to one of the possible experiment modes and were all presented with passages sampled from the same set, for which they were tasked with writing and answering five questions. All collected questions were then validated for correctness by a separate group of crowdworkers. We collect three validations per question and use this information, along with manual verification of a subset of the annotated examples, to maintain a high level of quality and remove examples from workers with less than an 80% validity rate. We calculate the reliability of agreement between validators using Fleiss’ kappa at 0.46. Workers were provided an additional bonus for each example validated as having successfully fooled the model in the adversarial data collection settings. In total,

²github.com/facebookresearch/Mephisto

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

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

1,931 workers participated in the study, with 1,559 contributing to the final datasets. We also continuously validate both annotators and validators based on signals such as repetitiveness, agreement, and manual checks.

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

- i. *median time per example* as a measure of annotation efficiency and where a lower time taken is better;
- ii. *validated Model Error Rate (vMER)* (Bartolo et al., 2021) which evaluates the effectiveness of annotators at generating valid question-answer pairs that the QA model in the loop fails to answer correctly;
- iii. *median time per validated model-fooling example* which serves as a single metric incorporating both method efficiency and effectiveness and thus provides a convenient metric for comparison across the various experimental settings; and
- iv. *downstream effectiveness* in which we evaluate the performance (by word-overlap F_1 score) of a QA model trained on the data collected in each of the experimental modes on the standard SQuAD1.1 benchmark, on the AdversarialQA benchmark, and in terms of domain generalisation ability on the MRQA (Fisch et al., 2019) dev sets.

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

4 Results

Our study allows us to perform a thorough investigation into both the efficiency and effectiveness of

the different data annotation methodologies. It also allows us to build on work investigating the various differences between standard and adversarial data collection (Kaushik et al., 2021).

4.1 Standard versus Adversarial Data Collection

The standard and adversarial data collection settings we use as baselines do not make use of GAAs, and are designed to replicate the SQuAD1.1 (Rajpurkar et al., 2016) and AdversarialQA (Bartolo et al., 2020) annotation setups as closely as possible. However, in contrast to AdversarialQA, our setting only provides annotators with a financial incentive to *try* to beat the model in the loop through the use of a bonus, and does not restrict annotators to only submitting model-fooling examples.

The results, shown in Table 1, highlight the differences between the two annotation approaches. As expected, standard data collection is more efficient in terms of the time taken per example, as there is no requirement for annotators to make any effort to try to beat a model. However, the efficiency differences are not as large as seen in settings where annotators *have* to submit model-fooling examples (Bartolo et al., 2020).

We also find considerable benefits from adversarial data collection in terms of the validated model error rate and subsequent downstream performance. As observed by Bartolo et al. (2020), adversarial data collection is more effective on adversarial test sets and aids domain generalisation, with slight performance degradation in the standard evaluation setting, which may be mitigated by increasing the amount of training data or combining with non-adversarial training data (for detailed results, refer to Appendix B). The combined performance across evaluation settings is considerably higher for adversarial data collection.

We note that the training data sizes in both these experimental settings are relatively small, and the benefits of adversarial data collection have been

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

Table 2: Results for the investigation into supporting standard data collection using GAAs. Since this setting assumes no access to adversarially-sourced data, we use a generative model trained only on questions from SQuAD1.1. There is no adversarial QA model in the loop in this setting.

shown to be more pronounced in the low data regime, likely due to increased example diversity. Furthermore, while the passages used in this study are sourced from Wikipedia, there may exist characteristic differences between these and the passages used in SQuAD.

We also observe considerably lower (i.e., better) adversarial human evaluation vMER scores achieved for our synthetically-augmented ELECTRA_{Large} model-in-the-loop compared to the 8.8% reported for RoBERTa_{Large} by Bartolo et al. (2021). We hypothesise that this is primarily due to two factors: the improved robustness of ELECTRA in comparison to RoBERTa, and more tightly-controlled example validation. For further evidence of the improved adversarial robustness of ELECTRA, refer to Appendix C.

4.2 Improving Standard Data Collection

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

We therefore use a GAA trained on SQuAD1.1, and investigate the three sampling techniques namely: likelihood, adversarial, and uncertainty sampling. Results are shown in Table 2. We find that using a GAA with likelihood sampling considerably improves the efficiency of the annotation process in comparison to the standard data collection baseline in Table 1. It also gives comparable vMER results and downstream QA performance.

Furthermore, both the adversarial and uncertainty sampling strategies prove effective. While the reduction in time taken per example is not as substantial as for standard likelihood sampling, and is comparable to the standard data collection baseline, the vMER – an indicator of the diversity of the collected training data – is substantially improved and outperforms the adversarial data collection baseline. The downstream results are also promising, providing slight improvements over the

standard data collection setting, particularly with regards to domain generalisation. They start to make progress towards the values obtained for the adversarial data collection baseline although, despite the improved vMER, overall downstream performance is considerably higher in the adversarial data collection setting.

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

4.3 Improving Adversarial Data Collection

Following the efficiency gains observed for standard data collection, we investigate whether it is possible for GAAs to provide further improvements over adversarial data collection. As for the previous experiments, we investigate GAAs trained on three different datasets: SQuAD1.1, AdversarialQA, and the combination of both. We combine each of these with the three previously discussed sampling strategies resulting in nine different experimental settings. Results are shown in Table 3.

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

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

Table 3: Results for the investigation into supporting adversarial data collection using GAAs. We investigate three different GAA training dataset sources, and three sampling strategies. The adversarial QA model used in the annotation loop is identical for all settings.

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

Table 4: Results for the investigation into supporting adversarial data collection using GAAs equipped with answer prompting. We investigate two different GAA training dataset sources, and three sampling strategies. The adversarial QA model-in-the-loop is identical for all settings.

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

In terms of downstream performance, there is no clear best option, but the best settings consistently outperform the adversarial data collection baseline on the most challenging examples ($\mathcal{D}_{\text{BERT}}$ and $\mathcal{D}_{\text{RoBERTa}}$) while providing comparable results in the other evaluation settings. Surprisingly, we find that various settings, particularly those involving a SQuAD-trained GAA can provide performance gains over the standard data collection baseline on SQuAD1.1. We also observe that a SQuAD-trained GAA with uncertainty sampling gives best performance on the less challenging evaluation sets, while an AdversarialQA-trained GAA gives best performance on the evaluation datasets collected using a more performant adversary. This is also in line with the observations made by Bartolo et al. (2020) showing a distributional shift in question type and complexity with an increasingly stronger model-in-the-loop.

The general takeaway in terms of the ideal experimental setting from the perspective of down-

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

4.4 Investigating Answer Prompting

The settings explored in the previous sections focus on investigating the effects of assisting free-text generation of the questions using GAAs. However, the QA crowdsourcing setting also involves annotation of answer spans, which we also explore in the search for efficiency gains. Here, we explore GAAs trained on datasets with adversarially-sourced components and the same three sampling strategies as previously (likelihood, uncertainty and adversarial), while additionally providing annotators with an answer suggestion.

In essence, this is similar to an answer and question validation setting, with the difference that annotators have the ability to freely modify both an-

swer and question, or request additional suggestions. Results for our experiments involving answer assistance are shown in Table 4.

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

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

5 Annotator Interaction with GAAs

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

Results, shown in Table 5, indicate that annotators query the GAA less frequently when being shown simpler prompts i.e. those obtained using a GAA trained on non-adversarially sourced examples, or selected using likelihood sampling which tends to provide higher quality and less complex generated texts. We also find that annotators query the GAA more frequently when an answer prompt is also provided. We believe that this can be attributed to the fact that the answer and question prompt setting is more similar to a validation work-

Feature	Setting	Avg. #Generations per Example
GAA Training	SQuAD	0.69
	AdversarialQA	0.86
	Combined	0.83
Sampling	<i>Likelihood</i>	0.67
	<i>Adversarial</i>	0.87
	<i>Uncertainty</i>	0.83
Answer Prompt?	\times	0.73
	\checkmark	0.91

Table 5: Results showing how often annotators query the GAA for different experimental settings.

flow, allowing annotators to generate prompts until a satisfactory one is found.

6 Discussion and Conclusion

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

We find that GAAs are beneficial in both the standard and adversarial data collection settings. In the standard data collection setting, and under the assumption of no access to adversarially-collected data, GAAs with prompts sampled based on likelihood provide annotation speed-ups, while prompts sampled by adversarial performance or uncertainty metrics provide benefits to both the model error rates on the collected data as well as subsequent downstream QA performance. We find that while GAAs are effective for improving standard data collection, we still do not approach the performance obtained when using adversarial data collection.

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

We find that the ideal annotation setting differs for different intended evaluations, with an uncertainty-sampled GAA trained on data that was not adversarially-collected providing best perfor-

mance on simpler questions, while a GAA trained on adversarially-collected data provides best downstream performance on more challenging evaluation sets. However, we also find that combining with a small sample of SQuAD training examples can boost performance on these less-challenging questions, and that in this setting a likelihood-sampled adversarially-trained GAA consistently provides the best results.

In terms of efficiency, we see annotation speed-ups over baseline of 34.4% for standard data collection and 37.4% for adversarial data collection. In terms of effectiveness, we see over a 5x improvement in vMER over adversarial data collection, along with downstream performance gains. We improve over standard data collection on SQuAD_{dev} by up to 0.8F₁ and improve over adversarial data collection by up to 6.1F₁ on $\mathcal{D}_{\text{BERT}}$, and 3.8F₁ on $\mathcal{D}_{\text{RoBERTa}}$. Furthermore, we see benefits in domain generalisation over standard data collection, and show that annotators interact with the GAA more frequently when it has been trained on adversarially-collected data, is sampled based on adversarial or uncertainty feedback, and also provides answer prompts.

While our analysis is limited by the size of the collected data, we believe that GAAs can help drive further innovation into improved data collection methodologies based on these findings. We hope that our analysis of various aspects of GAA incorporation into the annotation pipeline and the interactions between annotators and multiple models in the loop can help inform future work exploring broader aspects of GAA use, such as for other NLP tasks or for larger scale annotation efforts.

7 Ethical Considerations

We collect a training datasets as a part of the analysis in this work. The passages are sourced from Wikipedia through KILT. As described in the main text, our incentive structure is designed to ensure that crowdworkers were fairly compensated. Our datasets focus on the English language. As this data is not collected for the purpose of designing NLP applications, we do not foresee any risks associated with the use of this data.

Acknowledgements

The authors would like to thank the Dynabench team for their feedback and continuous support.

References

- Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. [Synthetic QA corpora generation with roundtrip consistency](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.
- Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. [Beat the AI: Investigating Adversarial Human Annotation for Reading Comprehension](#). *Transactions of the Association for Computational Linguistics*, 8:662–678.
- Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. [Improving Question Answering Model Robustness with Synthetic Adversarial Data Generation](#).
- Samuel R Bowman and George E Dahl. 2021. What will it take to fix benchmarking in natural language understanding? *arXiv preprint arXiv:2104.02145*.
- Samuel R. Bowman, Jennimaria Palomaki, Livio Baldini Soares, and Emily Pitler. 2020. [New protocols and negative results for textual entailment data collection](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8203–8214, Online. Association for Computational Linguistics.
- Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. [A thorough examination of the CNN/Daily Mail reading comprehension task](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany. Association for Computational Linguistics.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [Electra: Pre-training text encoders as discriminators rather than generators](#). In *International Conference on Learning Representations*.
- Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. [Build it break it fix it for dialogue safety: Robustness from adversarial human attack](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4537–4546, Hong Kong, China. Association for Computational Linguistics.
- Xinya Du and Claire Cardie. 2018. [Harvesting paragraph-level question-answer pairs from Wikipedia](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917, Melbourne, Australia. Association for Computational Linguistics.

- Xinya Du, Junru Shao, and Claire Cardie. 2017. [Learning to ask: Neural question generation for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.
- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. [DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.
- Allyson Ettinger, Sudha Rao, Hal Daumé III, and Emily M. Bender. 2017. [Towards linguistically generalizable NLP systems: A workshop and shared task](#). *CoRR*, abs/1711.01505.
- Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunso Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of 2nd Machine Reading for Reading Comprehension (MRQA) Workshop at EMNLP*.
- Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. [Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166, Hong Kong, China. Association for Computational Linguistics.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. [Annotation artifacts in natural language inference data](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.
- Robin Jia and Percy Liang. 2017. [Adversarial examples for evaluating reading comprehension systems](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.
- Divyansh Kaushik, Douwe Kiela, Zachary C. Lipton, and Wen-tau Yih. 2021. [On the efficacy of adversarial data collection for question answering: Results from a large-scale randomized study](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6618–6633, Online. Association for Computational Linguistics.
- Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. [Dynabench: Rethinking benchmarking in NLP](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.
- David D. Lewis and William A. Gale. 1994. [A sequential algorithm for training text classifiers](#). In *SIGIR*, pages 3–12. ACM/Springer.
- Mike Lewis and Angela Fan. 2019. [Generative question answering: Learning to answer the whole question](#). In *International Conference on Learning Representations*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. [PAQ: 65 million probably-asked questions and what you can do with them](#). *arXiv preprint arXiv:2102.07033*.
- Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. [Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.
- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. [Adversarial NLI: A new benchmark for natural language understanding](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. [fairseq: A Fast, Extensible Toolkit for Sequence Modeling](#). In *Proceedings of NAACL-HLT 2019: Demonstrations*.

- Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. [KILT: a benchmark for knowledge intensive language tasks](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.
- 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 arXiv:2111.08181*.
- Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2021. [DynaSent: A dynamic benchmark for sentiment analysis](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2388–2404, Online. Association for Computational Linguistics.
- Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. [Training question answering models from synthetic data](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online. Association for Computational Linguistics.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Alexis Ross, Tongshuang Wu, Hao Peng, Matthew E. Peters, and Matt Gardner. 2021. Tailor: Generating and perturbing text with semantic controls. *arXiv preprint arXiv:2107.07150*.
- Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. 2017. [The effect of different writing tasks on linguistic style: A case study of the ROC story cloze task](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 15–25, Vancouver, Canada. Association for Computational Linguistics.
- Burr Settles. 2009. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.
- Yoshimasa Tsuruoka, Jun’ichi Tsujii, and Sophia Ananiadou. 2008. [Accelerating the annotation of sparse named entities by dynamic sentence selection](#). In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 30–37, Columbus, Ohio. Association for Computational Linguistics.
- Clara Vania, Phu Mon Htut, William Huang, Dhara Mungra, Richard Yuanzhe Pang, Jason Phang, Haokun Liu, Kyunghyun Cho, and Samuel R. Bowman. 2021. [Comparing test sets with item response theory](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1141–1158, Online. Association for Computational Linguistics.
- Eric Wallace, Adina Williams, Robin Jia, and Douwe Kiela. 2021. Analyzing dynamic adversarial training data in the limit. *arXiv preprint arXiv:2110.08514*.
- Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. [Making neural QA as simple as possible but not simpler](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, Vancouver, Canada. Association for Computational Linguistics.
- Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. [Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6707–6723, Online. Association for Computational Linguistics.
- Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. [Generative data augmentation for commonsense reasoning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025, Online. Association for Computational Linguistics.
- Zhilin Yang, Saizheng Zhang, Jack Urbanek, Will Feng, Alexander H Miller, Arthur Szlam, Douwe Kiela, and Jason Weston. 2017. Mastering the dungeon: Grounded language learning by mechanical turker descent. *arXiv preprint arXiv:1711.07950*.
- Dani Yogatama, Cyprien de Masson d’Autume, Jerome T. Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and evaluating general linguistic intelligence. *ArXiv*, abs/1901.11373.
- Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. [Paragraph-level neural question generation with maxout pointer and gated self-attention networks](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.

A Breakdown of MRQA Results

Table 6 shows the breakdown of results on the 12 MRQA in- and out- of domain evaluation sets.

B Combining with SQuAD1.1

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

The baseline results in Table 7 show that this results in similar performance, if slightly improved on the SQuAD dev set, when using some adversarially-collected data. We also show the results for the other experimental settings in Tables 8, 9 and 10, noting very similar performance variation between settings as those reported earlier.

C Adversarial Robustness of ELECTRA and RoBERTa

Table 11 shows adversarial robustness performance evaluated on the AddSent and AddOneSent evaluation datasets introduced by Jia and Liang (2017). We observe that even when trained only on SQuAD1.1, ELECTRA performs considerably better than RoBERTa in this setting, suggesting that it is substantially more robust “out of the box”.

D Computational Resources

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

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

Table 6: Result breakdown for all twenty experiment modes on the MRQA evaluation sets.

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

Table 7: Baseline results comparing standard and adversarial data collection. Downstream evaluation is measured by training an ELECTRA_{Large} QA model on each of the collected datasets combined with 2k SQuAD training examples (for a total of 4k examples) and evaluating F₁ scores on the SQuAD1.1 dev set, the AdversarialQA test sets, and the MRQA dev sets for domain generalisation.

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

Table 8: Results for the investigation into supporting standard data collection using GAAs when combining with 2k SQuAD training examples. There is no adversarial QA model in the loop in this setting.

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

Table 9: Results for the investigation into supporting adversarial data collection using GAAs when combining with 2k SQuAD training examples. We investigate three different GAA training dataset sources, and three sampling strategies. The adversarial QA model used in the annotation loop is identical for all settings.

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

Table 10: Results for the investigation into supporting adversarial data collection using GAAs equipped with answer prompting when combining with 2k SQuAD training examples. We investigate two different GAA training dataset sources, and three sampling strategies. The adversarial QA model-in-the-loop is identical for all settings.

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

Table 11: Word-overlap F_1 results for BERT, RoBERTa, and ELECTRA on the SQuAD1.1 dev set and the AddSent and AddOneSent adversarial evaluation sets (Jia and Liang, 2017).