Retrieval-guided Counterfactual Generation for QA

Anonymous ACL submission

Abstract

Deep NLP models have been shown to be brittle to input perturbations. Recent work has shown that data augmentation using counterfactuals — i.e. minimally perturbed inputs — can help ameliorate this weakness. We focus on the task of creating counterfactuals for question answering, which presents unique challenges related to world knowledge, semantic diversity, and answerability. To address these challenges, we develop a ***R**etrieve-**G**enerate-**F**ilter* (RGF) technique to create counterfactual evaluation and training data with minimal human supervision. Using an open-domain QA framework and question generation model trained on original task data, we create counterfactuals that are fluent, semantically diverse, and automatically labeled. Data augmentation with RGF counterfactuals improves performance on out-of-domain and challenging evaluation sets over and above existing methods, in both the reading comprehension and open-domain QA settings. Moreover, we find that RGF data leads to significant improvements in a model's robustness to local perturbations.[1]

1 Introduction

Models for natural language understanding (NLU) may outperform humans on standard benchmarks, yet still often perform poorly under a multitude of distributional shifts (Jia and Liang (2017); Naik et al. (2018); McCoy et al. (2019), *inter alia*) due to over-reliance on spurious correlations or dataset artefacts. Recent work (Kaushik et al., 2020; Gardner et al., 2020) proposes the construction of *contrast* or *counterfactual* data — minimal yet meaningful perturbations to test examples that are created by humans to flip the task label — to expose gaps in a model's local decision boundaries. For instance, perturbing the movie review *"A real stinker, one out of ten!"* to *"A real classic, ten out of ten!"* changes its sentiment label. Kaushik et al. (2020, 2021); Wu et al. (2021a); Geva et al. (2021) show that augmenting with counterfactual data (CDA) improves out-of-domain generalization and robustness to small input perturbations. Consequently, several techniques have been proposed for the automatic generation of counterfactual data for several downstream tasks (Wu et al., 2021a; Ross et al., 2021b,a; Bitton et al., 2021; Geva et al., 2021; Asai and Hajishirzi, 2020; Mille et al., 2021).

[1] Code is available at https://anonymous/to/be/released

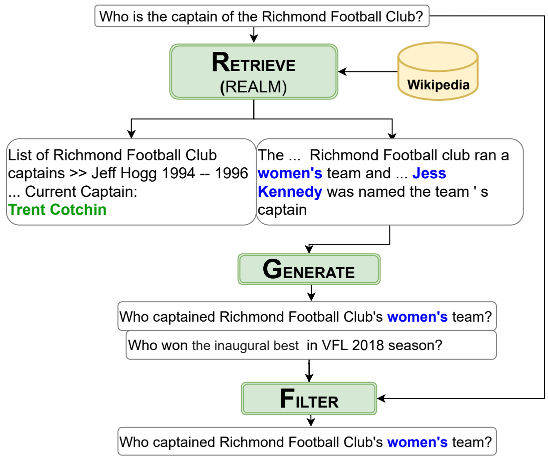

Figure 1: *Retrieve-Generate-Filter* to generate counterfactual queries for Natural Question (Kwiatkowski et al., 2019) using an open-domain retrieval system, question generation and post-hoc filtering.

In this paper, we focus on counterfactuals for question answering, in both the reading comprehension and open-domain settings (e.g. Rajpurkar et al., 2016; Kwiatkowski et al., 2019). Model inputs consist of a question and optionally a context passage, and the target is a short answer span. We define counterfactuals in this setting to be distinct but semantically proximal questions and (optionally) contexts. Unlike in classification, where the label space is fixed, the set of alternate labels, and hence alternate questions in QA is diverse, open-ended, and instance-specific. Exploring the semantic neighborhood around a particular question often requires world knowledge. For example, going from *"Who is the captain of the Richmond Football Club"* to *"Who captained Richmond's women's team?"* as in Figure 1 requires knowledge about the club's alternate teams, and the perturbation *"Who was the captain of RFC in 1998?"* requires knowledge about the time-sensitive nature of the original question. In the absence of such knowledge, otherwise reasonable edits — such as *"Who captained the club in 2050?"* — can result in false premises or unanswerable questions.

We develop a simple yet effective technique to address these challenges: ***R**etrieve, **G**enerate, and **F**ilter* (RGF; Figure 1). We use the near-misses of a retrieve-and-read QA model to propose alternate contexts and answers which are closely related to — but semantically distinct from — the original question. We then use a sequence-to-sequence question generation model (Alberti et al., 2019) to generate corresponding questions to these passages and answers. This results in fully-labeled examples, which can be used directly to augment training data or filtered post-hoc for analysis.

While our method requires no supervised inputs besides the original task training data, it is able to generate highly diverse counterfactuals covering a range of semantic phenomena (§4), including many transformation types which existing methods generate through heuristics (Dua et al., 2021), meaning representations (Ross et al., 2021b; Geva et al., 2021) or human generation (Bartolo et al., 2020; Gardner et al., 2020). Compared to alternative sources of synthetic data (§5.1), training augmented with RGF data leads to increased performance on a variety of settings (§5.2, §5.3), including out-of-domain (Fisch et al., 2019) and contrast evaluation sets (Bartolo et al., 2020; Gardner et al., 2020), while maintaining in-domain performance. Additionally, we introduce a measure of *pairwise consistency*, and show that RGF leads to significant improvements in model robustness to a range of local perturbations (§6).

2 Related Work

2.1 Counterfactual Generation

There has been considerable interest in developing challenging evaluation sets for NLU that evaluate models on a wide variety of counterfactual input perturbations. Gardner et al. (2020); Khashabi et al. (2020); Kaushik et al. (2020); Ribeiro et al. (2020) use humans to create these perturbations, optionally in an adversarial setting against a particular model (Bartolo et al., 2020). However, these methods can be expensive and difficult to scale.

This has led to an increased interest in creating *automatic* counterfactual data for evaluating out-of-distribution generalization (Bowman and Dahl, 2021) and for counterfactual data augmentation (Geva et al., 2021; Longpre et al., 2021). Some work focuses on using heuristics like swapping superlatives and nouns (Dua et al., 2021), changing gendered words (Webster et al., 2020), or targeting specific data splits (Finegan-Dollak and Verma, 2020). More recent work has focused on using meaning representation frameworks and structured control codes, including grammar formalisms (Li et al., 2020), semantic role labeling (Ross et al., 2021b), structured image representations like scene graphs (Bitton et al., 2021), and query decompositions in multi-hop reasoning datasets (Geva et al., 2021). Ye et al. (2021) and Longpre et al. (2021) perturb contexts instead of questions by swapping out all mentions of a named entity. The change in label can be derived heuristically or requires a round of human re-labeling of the data. These may also be difficult to apply to tasks like Natural Questions (Kwiatkowski et al., 2019), where pre-defined schemas can have difficulty covering the range of semantic perturbations that may be of interest.

2.2 Data Augmentation

Non-counterfactual data augmentation methods for QA, where the synthetic examples are not paired with the original data, have shown only weak improvements to robustness and out-of-domain generalization (Bartolo et al., 2021; Lewis et al., 2021). Moreover, Joshi and He (2021) find that methods that limit the structural and semantic space of perturbations can potentially hurt generalization to other types of transformations. This problem is exacerbated in the question answering scenario where there can be multiple semantic dimensions to edit. Our method attempts to address this by targeting a broad range of semantic phenomena, thus reducing the chance for the augmented model to overfit.

3 RGF: Counterfactuals for Information-seeking Queries

A counterfactual is commonly defined as a perturbed version of an input which alters a single latent variable while holding other aspects unchanged. While many existing works focus on label-preserving transformations, in this work we explore the much broader space of transformations which allow the label to change as well. For question-answering, we take as input triples (q, c, a) consisting of the question, context passage, and short answer, and produce counterfactual triples (q', c', a') where $a' \neq a$. This setting poses some unique challenges, such as the need for background knowledge to identify relevant semantic dimensions, ensuring sufficient semantic diversity in question edits, and avoiding questions with false premises or no viable answers. Ensuring (or characterizing) minimality can also be a challenge, as small changes to surface form can lead to significant semantic changes, and vice-versa. We introduce a generalized paradigm — ***R**etrieve, **G**enerate and **F**ilter* — to tackle these challenges.

3.1 Overview of RGF

An outline of the RGF method is given in Figure 1. Given an input example $x = (q, c, a)$ consisting of a question, a context paragraph, and the corresponding answer, RGF generates a set of new examples $N(x) = \{(q'_1, c'_1, a'_1), (q'_2, c'_2, a'_2), \ldots\}$ from the local neighborhood around x. We first use an open-domain retrieve-and-read model to retrieve alternate contexts c' and answers a' where $a \neq a'$. As near-misses for a task model, these candidates (c', a') are closely related to the original target (c, a) but often differ along interesting, latent semantic dimensions (Figure 2) in their relation to the original question, context, and answer. We then use a sequence-to-sequence question generation model to generate new questions q' from the context and answer candidates (c', a'). This yields triples (q', c', a') which are fully labeled, avoiding the problem of unanswerable or false-premise questions.

Compared to methods that rely on a curated set of minimal edits (e.g. Wu et al., 2021b; Ross et al., 2021b), our method admits the use of alternative contexts[2] $c' \neq c$, and we do not explicitly constrain our triples to be *minimal* counterfactuals during the generation step. Instead, we use post-hoc filtering to reduce noise, select minimal candidates, or select for specific semantic phenomena based on the relation between q and q'. This allows us to explore a significantly more diverse set of counterfactual questions q' (§C.1), capturing relations that may not be represented in the original context c.

We give a description of each component of RGF below; additional implementation details are provided in Appendix A.

3.2 Retrieval

We use REALM retrieve-and-read model of (Guu et al., 2020). REALM consists of consists of a BERT-based bi-encoder for dense retrieval, a dense index of Wikipedia passages, and a BERT-based answer-span extraction model for reading comprehension, all fine-tuned on Natural Questions (NQ; Kwiatkowski et al., 2019). Given a question q, REALM outputs a ranked list of contexts and answers within those contexts: $\{(c'_1, a'_1), (c'_2, a'_2), \ldots (c'_k, a'_k)\}$. These alternate contexts and answers provide relevant yet diverse background information to construct counterfactual questions. For instance, in Figure 1, the question *"Who is the captain of the Richmond Football Club"* with answer *"Trent Cotchin"* also returns other contexts with alternate answers like *"Jeff Hogg"* (q' =*"Who captained the team in 1994"*), and *"Steve Morris"* (q' =*"Who captained the reserve team in the VFL league"*). Retrieved contexts can also capture information about closely related or ambiguous entities. For instance, the question *"who wrote the treasure of the sierra madre"* retrieves passages about the original book *Sierra Madre*, its movie adaptation, and a battle fought in the Sierra de las Cruces mountains. This background knowledge allows us to perform *contextualized* counterfactual generation, without needing to specify *a priori* the type of perturbation or semantic dimension. To focus on label-transforming counterfactuals, we retain all (c'_i, a'_i) where a'_i does not match any of the gold answers a from the original NQ example.

3.3 Question Generation

This component generates questions q' that correspond to the answer-context pairs (c', a'). We use a T5 (Raffel et al., 2020) model fine-tuned on (q, c, a) triples from Natural Questions, using context passages as input with the answer marked with special tokens. We use the trained model to generate questions $(q'_1, q'_2, \ldots q'_k)$ for each of the the retrieved set of alternate contexts and answers, $((c'_1, a'_1), (c'_2, a'_2), \ldots (c'_k, a'_k))$. For each (c'_i, a'_i),

[2]An alternative approach would be to make direct, targeted edits to the original context c. However, beyond a limited space of local substitutions (Longpre et al., 2021; Ye et al., 2021; Ross et al., 2021a) this is very difficult due to the need to model complex discourse and knowledge relations.

we use beam decoding to generate 15 different questions q'. We measure the fluency and correctness of generated questions in §4.

3.4 Filtering for Data Augmentation

Noise Filtering The question generation model can be noisy, resulting in a question that cannot be answered given c' or for which a' is an incorrect answer. Round-trip consistency (Alberti et al., 2019; Fang et al., 2020) uses an existing QA model to answer the generated questions, ensuring that the predicted answer is consistent with the target answer provided to the question generator. We use an ensemble of six T5-based reading-comprehension ($(q, c) \rightarrow a$) models, trained on Natural Questions using different random seeds, and keep any generated (q', c', a') triples where at least 5 of the 6 models agree on the answer. This discards about 5% of the generated data, although some noise still remains; see §4 for further discussion.

Filtering for Minimality Unlike prior work on generating counterfactual perturbations, we do not explicitly control for the type of semantic shift or perturbation in the generated questions. Instead, we use post-hoc filtering over generated questions q' to encourage minimality of perturbation. We define a filtering function $f(q, q')$ that categorizes the semantic shift or perturbation in q' with respect to q. One simple version of f is the word-level edit (Levenshtein) distance between q and q'. After applying ensemble-based noise filtering, for each original (q, c, a) triple we select the generated (q', c', a') with the smallest word-edit distance between q and q' such that $a \neq a'$. We use this simple heuristic to create large-scale *counterfactual training data* for augmentation experiments (§5). Over-generating potential counterfactuals based on latent dimensions identified in retrieval and using a simple filtering heuristic avoids biasing the model toward a narrow set of perturbation types (Joshi and He, 2021).

3.5 Semantic Filtering for Evaluation

To better understand the types of counterfactuals generated by RGF, we can apply additional filters based on query meaning representations to categorize counterfactual (q, q') pairs during evaluation. Meaning representations provide a way to decompose a question into semantic units and categorize (q, q') based on which of these units are perturbed. In this work, we employ the QED formalism for

Question from NQ
Original: who is the captain of richmond football club?
Predicate: who is the captain of X?
Reference Change
CF1: who is the captain of richmond's vfl reserve team?
Predicate: who is the captain of X?
Predicate Change
CF2: who wears number 9 for richmond football club?
Predicate: who wears Y for X?
Predicate and Reference Change
CF3: who did graham negate in the grand final last year?
Predicate: who did X negate in Y last year?

Table 1: Categorization of generated questions based on QED decomposition. The original reference *"Richmond football Club"* changes in CF1 and CF3. Predicate *"Who is the captain"* changes in CF2 and CF3.

explanations in question answering (Lamm et al., 2021). QED explanations segment the question into a predicate template and a set of reference phrases. For example, the question *"Who is captain of richmond football club"* decomposes into one question reference *"richmond football club"* and the predicate *"Who is captain of X"*. A few example questions and their QED decompositions are illustrated in Table 1.

We use these query decompositions to identify the relation between a counterfactual pair (q, q'). Concretely, we fine-tune a T5-based model on the QED dataset to perform explanation generation following the recipe of Lamm et al. (2021), and use this to identify predicates and references for the question from each (q, c, a) triple in the evaluation data. We use exact match between strings to identify reference changes. As predicates can often differ slightly in phrasing (*who captained* vs. *who is captain*), we take a predicate match to be a prefix matching with more than 10 characters. For instance, *"Who is the captain of Richmond's first ever women's team?"* has the same predicate as *"Who is the captain of the Richmond Football Club"*. We then filter generated questions into three perturbation categories — reference change, predicate change, or both.

4 Intrinsic Evaluation

Following desiderata from Wu et al. (2021a) and Ross et al. (2021b), we evaluate our RGF data along three qualitative evaluation measures: *fluency*, *correctness*, and *directionality*.

Fluency Fluency measures whether the generated text is grammatically correct and semantically

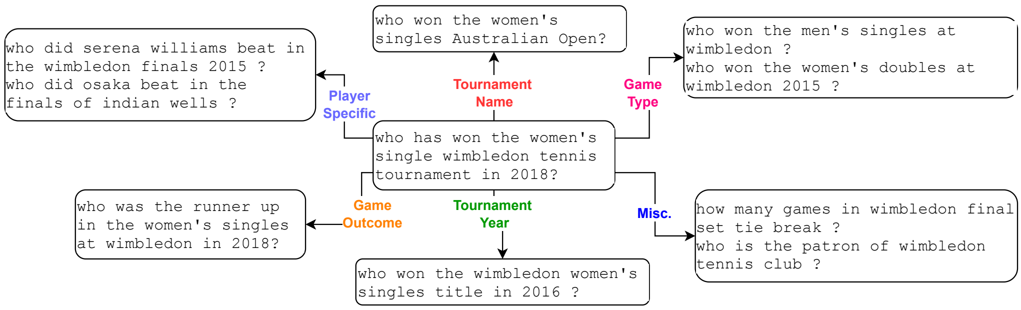

Figure 2: Context-specific semantic diversity of perturbations achieved by RGF on an NQ Question. The multiple latent semantic dimensions identified (arrows in the diagram) emergy from our retrieval-guided approach.

Semantic Change	Example (Original, Counterfactual)
Reference Change TAILOR (Ross et al., 2021b)	O: when did lebron_james join the Miami_Heat? C: When did lebron_james come into the league?
Predicate Change AmbigQA (Min et al., 2020b)	O: Who won the war between india and pakistan C: Who started the war between india and pakistan
Disambiguation AmbigQA (Min et al., 2020b)	O: When does walking dead season 8 start? C: When does walking dead season 8 second half start?
Negation Polyjuice (Wu et al., 2021a)	O: what religion observes the sabbath day C: what religion does not keep the sabbath day

Table 2: Patterns of semantic change between original queries (O) and RGF counterfactuals (C), corresponding to patterns explored by related works.

meaningful. Fluency is very high from RGF, as the generation step leverages a high-quality pretrained langauge model (T5). We manually annotate a subset of 100 generated questions, and find that 96% of these are fluent.

Correctness Correctness measures if the generated question q' and context, alternate answer pairs (c', a') are aligned i.e. the question is answerable given context c' and a' is that answer. We quantify correctness in the generated dataset by manually annotating a samples of 100 (q', c', a') triples (see Appendix B). The proportion of noise varies from 30% before noise filtering and 25% after noise filtering using an ensemble of models (§3.4).

Directionality/Semantic Diversity In Table 2, we show examples of semantic changes that occur in our data, including reference changes (50% of changes), predicate changes (30% of changes), negations, question expansions, disambiguations, and contractions. These cover many of the transformations found in prior work (Gardner et al., 2020; Ross et al., 2021b; Min et al., 2020b), but RGF is able to achieve these without the use of heuristic transformations or structured meaning representations. As shown in Figure 2, the types of relations are semantically rich and cover attributes relevant to each particular instance that would be difficult to capture with a globally-specified schema.

5 Data Augmentation

Unlike many counterfactual generation methods, RGF natively creates fully-labeled (q', c', a') examples which can be used directly for counterfactual data augmentation (CDA). We augment the original NQ training set with additional examples from RGF, shuffling all examples in training. We explore two experimental settings, reading comprehension (§5.2) and open-domain QA (§5.3), and compare RGF-augmented models to those trained only on NQ, as well as to alternative baselines for synthetic data generation. Additional training details for all models and baselines are included in Appendix A.

5.1 Baselines

In the abstract, our model for generating counterfactuals specifies a way of selecting contexts c' from original questions, and answers a' within those contexts, and a way of a generating questions q' from them. RGF uses a retrieval model to identify relevant contexts; here, we experiment with two baselines that use alternate ways to select c'.

Random Passage (Rand. Agen-Qgen) Here, c' is a randomly chosen paragraph from the Wikipedia index, with no explicit relation with the original question. This setting simulates generation from the original data distribution of Natural Questions. To ensure that the random sampling of Wikipedia paragraphs has a similar distribution, we employ the learned passage selection model from Lewis

Exact Match (RC)	Train Size	NQ	SQuAD	TriviaQA	HotpotQA	BioASQ	AQA	AmbigQA
Original NQ	90K	70.40	**80.22**	14.69	51.03	37.30	26.30	46.55
Gold Agen-Qgen	90K + 90K	70.60	74.64	13.24	45.59	31.98	20.50	43.45
Rand. Agen-Qgen	90K + 90K	**71.08**	76.78	13.87	45.26	33.64	22.50	42.04
RGF (REALM-Qgen)	90K + 90K	70.68	79.88	**16.99**	**53.41**	**44.88**	**28.20**	**47.61**

Table 3: Exact Match results for the reading comprehension task for in-domain NQ development set, out-of-domain datasets from MRQA 2019 Challenge (Fisch et al., 2019), Adversarial QA (Bartolo et al., 2020) and AmbigQA (Min et al., 2020b). RGF improves out-of-domain and challenge-set performance compared to other data augmentation baselines.

et al. (2021),[3] which was also used by Bartolo et al. (2021) in their closely related work on (non-counterfactual) data augmentation for the SQuAD dataset (Rajpurkar et al., 2016).

Gold Context (Gold Agen-Qgen) Here, c' is the passage c containing the original short answer a from the NQ training set. This baseline specifically ablates the retrieval component of RGF, testing whether the use of alternate passages leads to more diversity in the resulting counterfactual questions.

Answer Generation for Baselines For both the above baselines for context selection, we select spans in the new passage that are likely to be answers for a potential counterfactual question. We use a T5 (Raffel et al., 2020) model fine-tuned for question-independent answer selection $c \to a$ on NQ, and select the top 15 candidates from beam search. To avoid simply repeating the original question, we only retain answer candidates a' which do not match the original NQ answers a for that example. These alternate generated answer candidates and associated passages are then used for question generation and filtering as in RGF (§3.3).

5.2 Reading Comprehension (RC)

In the reading comprehension (RC) setting, the input consists of the question and context and the task is to identify an answer span in the context. Thus, we augment training with full triples (q', c', a') consisting of the retrieved passage c', generated and filtered question q', and alternate answer a'.

Experimental Setting We finetune a T5 (Raffel et al., 2020) model for reading comprehension, with input consisting of the question prepended to the context. We evaluate domain generalisation of our RC models on three evaluation sets from the MRQA 2019 Challenge (Fisch et al., 2019). We also measure performance on evaluation sets consisting of counterfactual or perturbed versions of RC datasets on Wikipedia, including SQuAD (Rajpurkar et al., 2016), AQA (adversarially-generated SQuAD questions; Bartolo et al., 2020), and human authored counterfactual examples (contrast sets; Gardner et al., 2020) from the QUOREF dataset (Dasigi et al., 2019). We also evaluate on the set of disambiguated queries in AmbigQA (Min et al., 2020b), which by construction are minimal edits to queries from the original NQ.

Results We report exact-match scores in Table 3; F1 scores follow a similar trend. We observe only limited improvements on the in-domain NQ development set, but we see significant improvements from CDA with RGF data in out-of-domain and challenge-set evaluations compared both to the original NQ model and the Gold and Random baselines. RGF improves by 1-2 EM points on most challenge sets, and up to 7 EM points on the BioASQ set compared to training on NQ only, while baselines often underperform the NQ-only model on these sets. Note that all three augmentation methods have similar proportion of noise (Appendix B), so CDA's benefits may be attributed to improving model's ability to learn more robust features for the task of reading comprehension and reduce reliance on spurious correlations. RGF's superior performance compared to the Gold Agen-Qgen baseline is especially interesting, since the latter also generates topically related questions. We observe that filtered RGF counterfactuals are more closely related to the original question compared to this baseline (Figure 5 in Appendix C), since q' is derived from a near-miss candidate (c', a') to answer the original q (S3.1).

5.3 Open-Domain Question Answering (OD)

In the open-domain (OD) setting, only the question is provided as input. The pair (q', a'), consisting

[3] https://github.com/facebookresearch/PAQ

Exact Match (OD)	Train Size	NQ	TriviaQA	AmbigQA	SQuAD v1.0	TREC
Original	90K	37.65	26.75	22.43	14.25	31.93
Gold Agen-Qgen	90K + 90K	37.86	27.02	23.65	15.01	32.94
Rand. Agen-Qgen	90K + 90K	37.45	29.87	24.13	14.55	26.89
RGF (REALM-Qgen)	90K + 90K	**39.11**	**32.32**	**26.98**	**16.94**	**33.61**

Table 4: Exact Match results on open-domain QA datasets (TriviaQA, AmbigQA, SQuAD and TREC) for data augmentation with RGF counterfactuals and baselines. Open-domain improvements are larger than in the RC setting, perhaps as the more difficult task benefits more from additional data.

of generated and filtered question q' and alternate answer a', is used for augmentation. Compared to the RC setting where passages change as well, here the edit distance filtering of §3.4 ensures the augmentation data represents minimal perturbations.

Experimental Setting We use the method and implementation from Guu et al. (2020) to finetune REALM on (q, a) pairs from NQ. End-to-end training of REALM updates both the reader model and the query-document encoders of the retriever module. We evaluate domain generalization on popular open-domain benchmarks: TriviaQA (Joshi et al., 2017), SQuAD (Rajpurkar et al., 2016), Curated TREC dataset (Min et al., 2021), and disambiguated queries from AmbigQA (Min et al., 2020b).

Results In the open-domain setting, we observe an improvement of 2 EM points over the original model even in the in-domain setting on Natural Questions (Table 4), while also improving significantly when compared to other data augmentation techniques. RGF improves over the next best baseline — Random Agen-Qgen — by up to 6 EM points (for TriviaQA). We hypothesize that data augmentation has more benefit in this setting, as the open-domain task is more difficult than reading comprehension, and counterfactual queries may help the model learn better query and document representations to improve retrieval.

6 Analysis

To better understand how CDA improves the model, we introduce a measure of local consistency (§6.1) to measure model robustness, and perform a stratified analysis (§6.2) to show the benefits of the semantic diversity available from RGF.

6.1 Local Robustness

Compared to synthetic data methods such as PAQ (Lewis et al., 2021), RGF generates counterfactual examples that are paired with the original inputs and concentrated in local neighborhoods around them (Figure 2). As such, we hypothesize that augmentation with this data should specifically improve local consistency, i.e. how the model behaves under small perturbations of the input.

Experimental Setting We explicitly measure how well a model's local behavior respects perturbations to input. Specifically, if a model $f : (q, c) \rightarrow a$ correctly answers q, how often does it also correctly answer q'? We define *pairwise consistency* as accuracy over the counterfactuals (q', a', c'), conditioned on correct predictions for the original examples:

$$\mathbb{C}(D) = E_D[f(q', c') = a' \mid f(q, c) = a]$$

To measure consistency, we construct validation sets consisting of paired examples $(q, c, a), (q', c', a')$: one original, and one counterfactual. We use QED to categorize our data, as described in §3.5. Specifically, we create two types of pairs: (a) a change in reference where question predicate remains fixed, and (b) a change in predicate, where the original reference(s) are preserved.[4] We create a clean evaluation set by first selecting RGF examples for predicate or reference change, then manually filtering the data to discard incorrect triples (§4) until we have 1000 evaluation pairs of each type (see Appendix B).

We also construct paired versions of AQA, AmbigQA, and the QUOREF contrast set. For AmbigQA, we pair two disambiguated questions and for the QUOREF contrast set, we pair original and human-authored counterfactuals. AQA consists of human-authored adversarial questions q' which are not explicitly paired with original questions; we create pairs by randomly selecting an original question q and a generated question q' from the same passage.

[4] We require that the new reference set is a superset of the original, since predicate changes can introduce additional reference slots (see CF2 in Table 1).

Consistency (RC)	Train Size	AQA	AmbigQA	QUOREF-C	RGF (Δ Ref)	RGF (Δ Pred)
Original NQ	90K	58.47	46.67	39.66	64.57	51.50
Gold Agen-Qgen	90K + 90K	59.27	50.23	42.83	44.62	38.10
Rand. Agen-Qgen	90K + 90K	55.45	49.06	41.93	60.77	48.53
RGF (REALM-Qgen)	90K + 90K	**63.29**	**51.61**	**46.42**	**76.36**	**64.98**
Effect of Perturbation Type						
RGF Δ Ref.	90K + 52K	62.55	**53.85**	41.34	**80.10**	62.33
RGF Δ Pred.	90K + 52K	**64.10**	47.45	**44.25**	74.96	**64.08**

Table 5: Results for pairwise consistency (§6.1) on reading comprehension, measured for datasets containing pairs of very similar questions. QUOREF-C refers to the QUOREF contrast set from (Gardner et al., 2020). RGF leads to better consistency in RC and open-domain settings (Appendix C.2). Results on effect of perturbation type during training (Δ Ref. and Δ Pred.) suggest that perturbation-bias does not degrade consistency over the original model.

Results Training with RGF data improves consistency by 12-14 points on the QED-filtered slices of RGF data, and 5-7 points on AQA, AmbigQA and QUOREF contrast (Table 5). The Gold Agen-Qgen baseline (which contains topically related queries about the same passage) also improves consistency over the original model compared to the Random Agen-Qgen baseline. Consistency improvements on AQA, AmbigQA and QUOREF are especially noteworthy, since they suggest an improvement in robustness to local perturbations that is independent of other confounding distributional similarities between training and evaluation data.

6.2 Effect of Perturbation Type

QED-based decomposition of queries allows for the creation of label-changing counterfactuals along *orthogonal* dimensions — a change of reference or predicate. We investigate whether training towards one type of change induces generalization bias, a detrimental effect which has been found in tasks like NLI (Joshi and He, 2021).

Experimental Setting We shard training examples into two categories based on whether q and q' have the same reference (predicate change) or same predicate (reference change), as defined in §3.5. We over-generate by starting with 20 (q', c', a') for each original training example to ensure that we find at least one q' that matches the criterion. We similarly evaluate on the paired evaluation sets from §6.1.

Results Results are shown for QED-filtered training in Table 5. Counterfactual perturbation of a specific kind (a predicate or a reference change) during augmentation does not hurt performance on another perturbation type compared to the baseline NQ model, which differs from the observations of Joshi and He (2021) on NLI. Furthermore, similar to the observations of Min et al. (2020a), augmenting with one type of perturbation has orthogonal benefits that improve model generalization on another perturbation type: augmenting with RGF (Δ Pred.) leads to significant improvement on RGF (Δ Ref.), and vice-versa. Overall, we observe that augmenting with predicate-change examples leads to greater improvements in local consistency compared to reference-change examples, except for on RGF (Δ Ref.) and on AmbigQA which contains a disproportionate number of reference-change pairs. Predicate-change examples may be more informative to the model, as reference changes can be modeled more easily by lexical matching within common context patterns.

7 Conclusion

Retrieve-Generate-Filter (RGF) creates counterfactual examples for question-answering which are semantically diverse, using knowledge from the passage context and a retrieval model to capture semantic changes that would be difficult to specify *a priori* with a global schema. The resulting examples are fully-labeled, and can be used directly for training augmentation or filtered using heuristics or meaning representations for analysis. We show that training with this data leads to improvements on open-domain QA, as well as on challenge sets, and leads to significant improvements in local robustness. While in this paper we focus on question answering, a task for which retrieval components are readily available, we note that the RGF paradigm is quite general and could potentially be applied to many other tasks with suitable choice of context and retrieval system.

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.

Akari Asai and Hannaneh Hajishirzi. 2020. Logic-guided data augmentation and regularization for consistent question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5642–5650, Online. 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. *arXiv preprint arXiv:2104.08678*.

Yonatan Bitton, Gabriel Stanovsky, Roy Schwartz, and Michael Elhadad. 2021. Automatic generation of contrast sets from scene graphs: Probing the compositional consistency of GQA. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 94–105, Online. Association for Computational Linguistics.

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

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5925–5932, Hong Kong, China. Association for Computational Linguistics.

Dheeru Dua, Pradeep Dasigi, Sameer Singh, and Matt Gardner. 2021. Learning with instance bundles for reading comprehension. *arXiv preprint arXiv:2104.08735*.

Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. 2020. Accelerating real-time question answering via question generation. *arXiv preprint arXiv:2009.05167*.

Catherine Finegan-Dollak and Ashish Verma. 2020. Layout-aware text representations harm clustering documents by type. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 60–65, Online. Association for Computational Linguistics.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Mor Geva, Tomer Wolfson, and Jonathan Berant. 2021. Break, perturb, build: Automatic perturbation of reasoning paths through question decomposition. *arXiv preprint arXiv:2107.13935*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

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.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Nitish Joshi and He He. 2021. An investigation of the (in) effectiveness of counterfactually augmented data. *arXiv preprint arXiv:2107.00753*.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. *International Conference on Learning Representations*.

Divyansh Kaushik, Douwe Kiela, Zachary C. Lipton, and Wen-tau Yih. 2021. On the efficacy of adversarial data collection for question answering: Results from a large-scale randomized study. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6618–6633, Online. Association for Computational Linguistics.

Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. 2020. More bang for your buck: Natural perturbation for robust question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 163–170, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. 2021. QED: A Framework and Dataset for Explanations in Question Answering. *Transactions of the Association for Computational Linguistics*, 9:790–806.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. 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*.

Chuanrong Li, Lin Shengshuo, Leo Z Liu, Xinyi Wu, Xuhui Zhou, and Shane Steinert-Threlkeld. 2020. Linguistically-informed transformations (lit): A method for automatically generating contrast sets. *arXiv preprint arXiv:2010.08580*.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. *arXiv preprint arXiv:2109.05052*.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2021. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1075–1088.

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.

Simon Mille, Thiago Castro Ferreira, Anya Belz, and Brian Davis. 2021. Another PASS: A reproduction study of the human evaluation of a football report generation system. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 286–292, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Junghyun Min, R Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020a. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352.

Sewon Min, Kenton Lee, Ming-Wei Chang, Kristina Toutanova, and Hannaneh Hajishirzi. 2021. Joint passage ranking for diverse multi-answer retrieval. *arXiv preprint arXiv:2104.08445*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020b. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Pranav Rajpurkar, 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.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Alexis Ross, Ana Marasović, and Matthew Peters. 2021a. Explaining NLP models via minimal contrastive editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852, Online. Association for Computational Linguistics.

Alexis Ross, Tongshuang Wu, Hao Peng, Matthew E Peters, and Matt Gardner. 2021b. Tailor: Generating and perturbing text with semantic controls. *arXiv preprint arXiv:2107.07150*.

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, Ed Chi, and Slav Petrov. 2020. Measuring and reducing gendered correlations in pre-trained models. *arXiv preprint arXiv:2010.06032*.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021a. 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.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. 2021b. Polyjuice: Automated, general-purpose counterfactual generation. *arXiv preprint arXiv:2101.00288*.

Xi Ye, Rohan Nair, and Greg Durrett. 2021. Evaluating explanations for reading comprehension with realistic counterfactuals. *arXiv preprint arXiv:2104.04515*.

A Model Training and Implementation Details

Below, we describe the details of different models trained in the RGF pipeline. For all T5 models, we use the pre-trained checkpoints from Raffel et al. (2020)[5].

Question Generation We use a T5-3B model fine-tuned on Natural Questions (NQ) dataset. We only train on the portion of the dataset that consists of gold short answers and an accompanying long answer evidence paragraph from Wikipedia. The input consists of the title of the Wikipedia article the passage is taken from, a separator ('»') and the passage. The short answer is enclosed in the passage using character sequences '« answer =' and '»' on left and right respectively. The output is the original NQ question. The input and output sequence lengths are restricted to be 640 and 256 respectively. We train the model for 20k steps with a learning rate of $2 \cdot 10^{-5}$, dropout 0.1, and batch size of 128. We decode with a beam size of 15, and take the top candidate as our generated question q'.

Answer Generation We use a T5-3B model trained on the same subset of Natural Questions (NQ) as question generation with same set of hyper-parameters and model size described above. The input consists of the title of the Wikipedia article the passage is taken from, a separator ('»') and the passage, while the output sequence is the short answer from NQ.

Reading Comprehension Model We model the task of span selection-based reading comprehension, i.e. identifying an answer span given question and passage, as a sequence-to-sequence problem. Input consists of the question, separator ('»'), and title of Wikipedia article, separator ('»') and passage. The answer format is simply one of the gold answer strings. The reading comprehension model is a T5-large model trained with batch size of 512 and learning rate $2 \cdot 10^{-4}$ for $20K$ steps.

Open-domain Question Answering model The open domain QA model is based on the implementation from (Lee et al., 2019), and initialized with the REALM checkpoint from (Guu et al., 2020)[6]. Both the retriever and reader are initialized from the BERT-base-uncased model. The query and document representations are 128 dimensional vectors. When finetuning, we use a learning rate of 10^{-5} and a batch size of 1 on a single Nvidia V100 GPU. We perform 2 epochs of fine-tuning for Natural Questions.

Noise Filtering We train 6 reading comprehension models based on the configurations above with different seed values for randomizing training dataset shuffling and optimizer initialization. We retain examples where more than 5 out of 6 models have the same answer for a question.

QED Training We use a T5-large model fine-tuned on the Natural Questions subset with QED annotations (Lamm et al., 2021).[7] We refer the reader to the QED paper for details on the linearization of explanations and inputs in the T5 model. Our model is fine-tuned with batch size of 512 and learning rate $2 \cdot 10^{-4}$ for 20k steps.

[5] https://github.com/google-research/text-to-text-transfer-transformer#released-model-checkpoints

[6] https://github.com/google-research/language/tree/master/language/realm

B Evaluation of Fluency and Noise

The authors sampled 300 examples of generated questions. To annotate for fluency, authors use the following rubric: Is the generated question grammatically well-formed barring non-standard spelling and capitalization of named entities. This noise annotation was done for RGF, as well as Gold Agen-Qgen and Random Agen-Qgen.

Data	Unfiltered	Filtered
RGF	29.8%	25.3%
Gold Agen-Qgen	27.9%	20.7%
Random Agen-Qgen	30.7%	28.3%

Table 6: Fraction of noise (incorrect (q', c', a')) in generated data, from 300 examples manually annotated by the authors.

Creation of paired data for counterfactual evaluation Once again, authors annotate for correctness of counterfactual RGF instances that are paired by reference or predicate, as described in §3.5. Filtering is done until 1000 examples are available under each category.

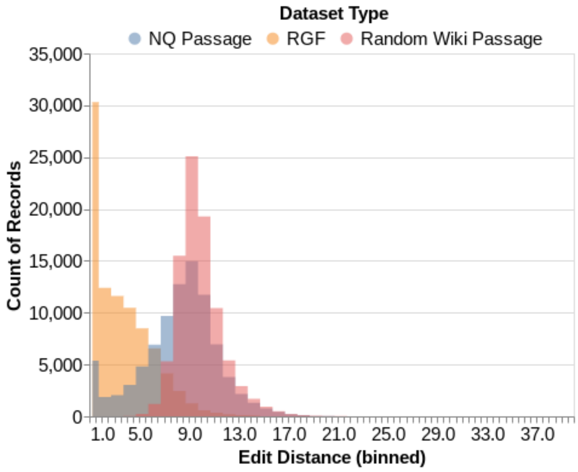

Figure 3: Distribution of edit distance between original q and counterfactual q' for RGF and other baselines for context selection. Note: For Random Wiki Passage, original and generated questions bear no relation to each other and are randomly paired.

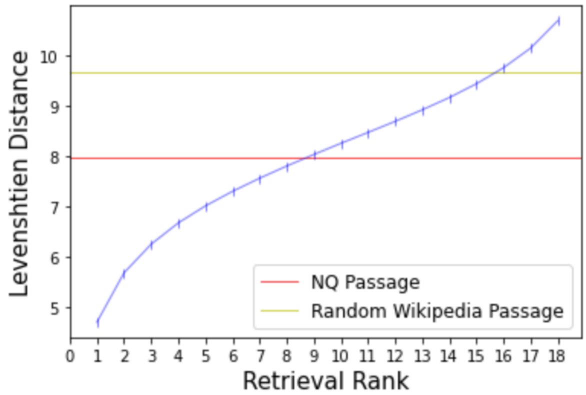

Figure 4: Plot of average edit distance between q, q' vs. retrieval rank r, where q' is generated from r^{th} passage, showing that edit distance and retrieval rank are monotonically related.

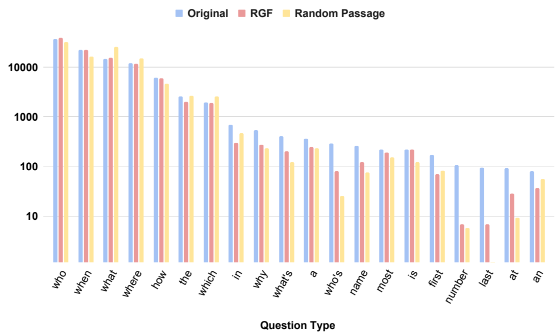

Figure 5: Distribution of top 20 question types for original NQ data, RGF counterfactuals and questions generated from random Wikipedia passage, indicating bias towards popular question types.

C Additional Experiments

C.1 Intrinsic Evaluation

In Figure 3, we compare distributions of the edit distance between the original and generated questions for questions generated by our approach, those generated with the gold evidence passage, and those generated from a random Wikipedia passage (§5). We find that RGF counterfactuals undergo minimal perturbations from the original question compared to questions that are generated from random Wikipedia paragraph. Surprisingly, this pattern also holds when compared to questions generated from gold NQ passages. We hypothesize that the set of alternate answers retrieved in our pipeline approach are semantically similar to the gold answer — same entity type, for instance. Random answer spans chosen from the *gold NQ passage* can result in significant semantic shifts in generated questions.

In Figure 4, we measure the relation between retrieval rank and edit-distance for RGF. For retrieval rank i, we plot average edit distance between the original question and counterfactual question that was generated using the ith passage and answer. We observe a monotonic relation between retrieval rank and edit distance (which we use for filtering our training data). We measure changes in the distribution of question type and predicate type.

Figure 5 indicates that counterfactual data exacerbates question-type bias. However, this bias exists in RGF as well as baselines.

C.2 Consistency for Open-Domain QA

In Table 7, we show results on evaluating consistency on paired datasets in the open-domain results, similar to the results shown in §6.1 in the Reading Comprehension setting.

C.3 Low-resource Transfer

Joshi and He (2021) show CDA to be most effective in the low-resource regime. To better understand the role that dataset size plays in CDA in the reading comprehension setting, we evaluate RGF in a cross-domain setting where only a small amount of training data is available.

Experimental Setting Since our approach depends on using an open-domain QA model and a question generation model trained on all Natural

[7] https://github.com/google-research-datasets/QED

Consistency (OD)	Train Size	AQA	AmbigQA	RGF Δ Ref.	RGF Δ Pred.
Original NQ	90K	16.58	13.33	25.12	11.23
Random Agen-Qgen	90K + 90K	15.80	20.00	27.94	17.16
RGF (REALM-Qgen)	90K + 90K	**17.66**	**28.57**	**31.77**	**19.81**

Table 7: Consistency Results for Open-domain QA.

Training Data	Train Size	BioASQ (Dev)	
		F1	EM
Original	1000	42.93	23.67
Orig. + RGF	500 + 500	41.72	23.01
Original	2000	**45.88**	25.80
Orig. + RGF	1000 + 1000	44.64	**26.80**

Table 8: Results on the reading comprehension task for Low Resource Transfer setting on BioASQ 2019 dataset. A model trained on 1000 gold BioASQ plus 1000 RGF examples performs nearly as well as a model trained on 2000 gold examples.

Consistency (RC)	Val 1-4	Val 5-10	Val > 10
Train 1-4	71.02	67.55	64.78
Train 5-10	68.89	**68.98**	63.92
Train >10	65.78	66.33	**65.33**
Train All	**72.34**	67.82	65.12

Table 9: Results on sharding training data based on edit distance between (q, q'). Training dataset size for each bin is 90k NQ + 167k generated. Once again, training with all RGF data robustly improves consistency across different amounts of perturbations.

Questions data, we instead experiment with a low-resource transfer setting on the BioASQ domain, which consists of questions on the biomedical domain. We use the domain-targeted retrieval model from (Ma et al., 2021), where synthetic question-passage relevance pairs generated over the PubMed corpus are used to train domain-specific retrieval without any in-domain supervision. We further fine-tune the question generation model trained on NQ on the limited amount of in-domain data, and use a checkpoint trained on NQ as an initialization to fine-tune the RC model for in-domain data. Details of our training approach for low-resource transfer can be found in Appendix A.

Results We observe significant improvements over the baseline model in the low resource setting for in-domain data (< 2000 examples), as shown in Table 8. Compared with the limited gains we see on the relatively high-resource NQ reading comprehension task, we find that on BioASQ, CDA with 1000 examples improves performance by 2% F1 and 3% exact match, performing nearly as well as a model trained on 2000 gold examples.

C.4 Effect of perturbation type

Experimental Setting For edit distance-based experiments, we shard training examples into three categories by binning word-level edit distance between q and q' into three ranges: 1–4, 5–10, and > 10. We similarly categorize RGF data generated for the NQ development set into the same categories. Evaluation sets for edit-distance experiments based were not manually noise filtered. We again report consistency on the reading comprehension model.

Results Similar to the observations for dataset sharding along QED annotations, when data is sharded by edit distance, we observe that using the full RGF data nearly matches the best performance from training on that shard, suggesting that CDA with the highly diverse RGF data can lead to improved consistency on a broad range of perturbation types.

D Semantic Diversity

Figure 6 includes more examples from Natural Questions, showing the counterfactual questions generated for different input questions by RGF.

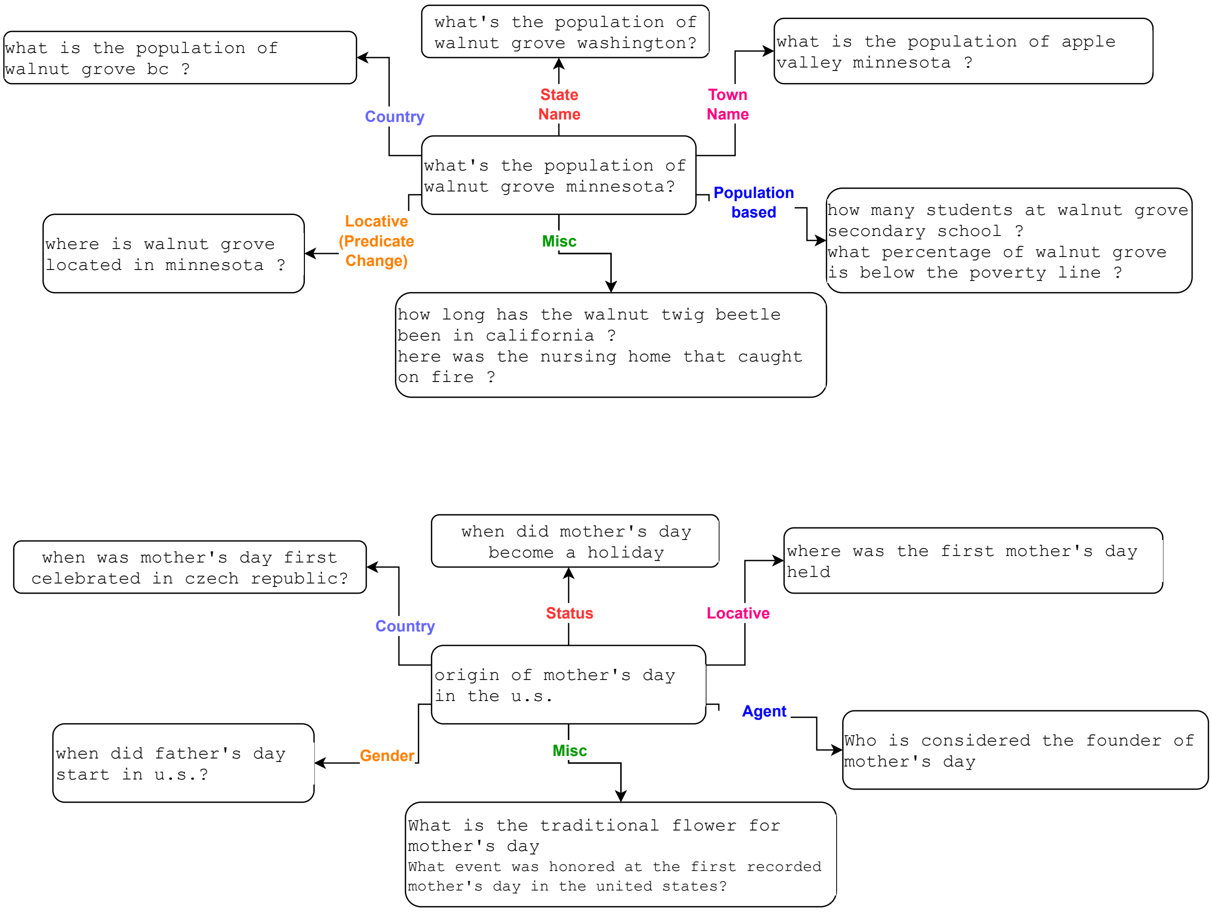

Figure 6: Context-specific semantic diversity of perturbations achieved by RGF on an NQ Question. The multiple latent semantic dimensions identified (arrows in the diagram) fall out of our retrieval-guided approach.