Semantic, Syntactic, Lexical: What Makes QA Augmentation Work in Limited Quantity?

Benedictus Kent Rachmat[1,2], Thomas Gerald[1], Takuya Nakamura[1], Zheng Zhang[2], Cyril Grouin[1]
[1]Université Paris-Saclay, CNRS, LISN, Orsay, France
[2]Embedded AI Lab, SLB, Clamart, France
{rachmat, gerald, nakamura, grouin}@lisn.fr, zhang@slb.com

Abstract

Data augmentation is a common fix in domains where training data is scarce or difficult to collect, such as specialized medical or any other domain specific applications. In question answering (QA), most studies report headline accuracy while saying little about the quality of the synthetic data. Here, quality goes beyond fluent rewording: augmented items must remain faithful to the supporting evidence and preserve the original answerability. We study three augmentation families *lexical*, *syntactic*, and *semantic* edits generated with LLaMA 3.1 70B, and analyze how these edits affect model behavior. To mirror low-resource settings, we focus on subsets of SQuADv2 (general) and PubMedQA (biomedical, domain specific). We report Exact Match (EM)/F1 alongside quality diagnostics, yielding a fuller picture than accuracy alone. Our results show that augmentation behaves differently across domains and scales. In SQuADv2, augmented variants maintain performance on par with baselines, showing that added diversity mostly does not harm model quality, whereas in PubMedQA semantic edits bring improvements under extreme scarcity and support stronger performance as supervision grows.

Keywords: Question Answering, Evaluation Metrics, Human Validation, Diversity, Synthetic Data, Domain Specific, Data Augmentation

1. Introduction

Data augmentation offers a straightforward promise to generate additional training examples to compensate for scarce annotations and improve model generalization. This promise has been realized in many NLP tasks, especially classification and text generation (Feng et al., 2021; Dai et al., 2025), where methods like paraphrasing and back-translation yield consistent gains (Sobrevilla Cabezudo et al., 2024). By exposing models to varied phrasing and diverse examples, augmentation can improve robustness, particularly in low-resource settings where labeled data is scarce. Large language models (LLMs) further enable synthetic corpus creation at scale, but raise a key question: what kind of linguistic variation do they actually produce, and which kind is most useful for downstream learning?

In question answering (QA), augmented examples must satisfy constraints beyond surface fluency, and the edits an augmentation system can apply to a question–answer pair fall into three linguistically motivated families. *Lexical* augmentation replaces individual words with synonyms or near-synonyms (e.g., *meet* → *encounter*), varying surface vocabulary while preserving meaning and structure. *Syntactic* augmentation restructures the sentence. For instance, converting active to passive voice, reordering clauses or changing grammatical form but not propositional content. *Semantic* augmentation goes further: given the same source passage, it generates an entirely new question targeting a different fact. These three levels mirror classical linguistic analysis (lexicon, syntax, semantics) and allow us to isolate which kind of variation is most beneficial for model training.

In contextual QA, the system must extract an answer span from a given passage or return "unanswerable" if none exists, so augmented examples must remain faithful to the passage and preserve answerability. Prior work has emphasized scale, as large synthetic sets often boost exact match and F1 (Alberti et al., 2019; Shakeri et al., 2020), yet without quality control, such gains may be misleading artifacts. We study this on two contrasting benchmarks: SQuADv2 (Rajpurkar et al., 2018), a general-domain dataset built from Wikipedia with both answerable and unanswerable questions, and PubMedQA (Jin et al., 2019), a biomedical dataset derived from PubMed abstracts requiring a categorical label (*yes*/*no*/*maybe*) and a free-text justification. Rather than using full training sets, we sample small subsets to mimic low-resource conditions, as truly low-resource QA datasets tend to be noisy and under-explored (Castelli et al., 2020; Jin et al., 2022); these well-established benchmarks provide a controlled testbed for studying how augmentation generalizes across domains and data scales.

Contributions. Our main contributions are:

- Cross-domain study: to our knowledge, the first systematic comparison of lexical, syntactic, and semantic augmentation across SQuADv2 (general) and PubMedQA (biomedical)

Proceedings of the Workshop on Structured Linguistic Data and Evaluation, pages 224–236

- Augmentation pipeline: we design a controlled setup where each augmentation type is generated and validated with GPT-4o, with a human audit confirming strengths and failure modes
- Scaling analysis: we reveal non-monotonic behavior across supervision scales, linking overfitting at mid-size subsets to LoRA dynamics
- Error taxonomy: we provide a structured taxonomy of augmentation failures, offering diagnostic tools for improving future QA augmentation

2. Related Work

Data augmentation for QA operates at different linguistic levels. Lexical methods such as synonym substitution expand vocabulary diversity (Wei and Zou, 2019). Syntactic approaches rephrase questions via structural changes like back-translation (Sennrich et al., 2016), improving robustness to phrasing (Yu et al., 2018). Semantic augmentation leverages generative models to create new QA pairs or richer paraphrases, yielding strong gains in benchmarks like SQuADv2 and PubMedQA (Alberti et al., 2019; Shakeri et al., 2020; Guo et al., 2023). While foundational studies showed improvements, later work highlighted mixed results: for example, back-translation sometimes fails to transfer across domains (Longpre et al., 2019).

LLM-based augmentation. Recent work has increasingly leveraged LLMs as data generators. Schmidt et al. (2024) show that LLM prompting generates high-quality synthetic QA data that improves few-shot performance. Chowdhury and Chadha (2024) demonstrate that LLM-generated augmentation improves distributional robustness across multiple QA datasets. Chan et al. (2024) categorize synthetic data strategies into answer augmentation, question rephrase, and new question generation, a taxonomy that parallels our lexical/syntactic/semantic distinction and show the optimal strategy depends on the seed-to-query ratio. In domain-specific settings, Khlaut et al. (2024) use GPT-4 to generate medical QA training data from textbooks, achieving competitive performance with smaller fine-tuned models.

Quality control and synthetic data risks. Scaling augmentation without fidelity may harm rather than help. Chen et al. (2024) identify that uniform format in synthetic QA pairs leads to pattern overfitting and distribution shifts. Pletenev et al. (2025) study knowledge packing in LoRA adapters on Llama-3.1-8B-Instruct, finding that biased synthetic training data causes regression to overrepresented answers. For quality validation, the LLM-as-a-Judge paradigm (Zheng et al., 2023; Gu et al., 2025) has shown that GPT-4 achieves $>$80% agreement with human experts, though position and verbosity biases persist.

Positioning. Our work integrates lexical, syntactic, and semantic edits in one framework with a validation loop, combining the systematic comparison of augmentation types with quality-aware filtering. Unlike prior work that isolates single strategies, we study their interactions across data scales and domains.

3. Methodology

3.1. Datasets

We evaluate on two extractive QA benchmarks: SQuADv2 (Rajpurkar et al., 2018), also known as SQuADRUn, a general-domain reading comprehension dataset that extends the original SQuAD (Rajpurkar et al., 2016) with unanswerable questions, and PubMedQA (Jin et al., 2019), a biomedical QA dataset derived from PubMed abstracts. This pairing lets us assess whether augmentation strategies transfer across domains and linguistic distributions. To simulate low-resource settings, we downsample each training set to $\frac{1}{32}$ of its original size as the base augmentation pool, then sample progressively smaller nested fractions ($\frac{1}{128} \subset \frac{1}{64} \subset \frac{1}{32} \subset \frac{1}{16}$). Test sets remain fixed (11,873 for SQuADv2, 1,000 for PubMedQA). Since neither benchmark provides an official dev set, we reserve 10% of the original training data as a validation pool, and consistently hold out 10% of each training portion for fine-tuning validation. For SQuADv2, we apply stratified sampling to preserve the answerable/unanswerable ratio. Each downsampled split therefore contains both answerable and unanswerable items; the unanswerable questions are carried into augmentation.

Fraction	SQuADv2	PubMedQA
$1/128$	1,015	1,648
$1/64$	2,031	3,296
$1/32$	4,062	6,593
$1/16$	8,125	13,187

Table 1: Training sample counts per supervision scale

To characterize the data geometry, we project samples into embedding space using Qwen3-Embedding-0.6B. As shown in Figure 1, the two corpora exhibit clear domain separation, though some SQuADv2 passages with biomedical content cluster near PubMedQA and vice versa, indicating the embedding model captures cross-domain se-

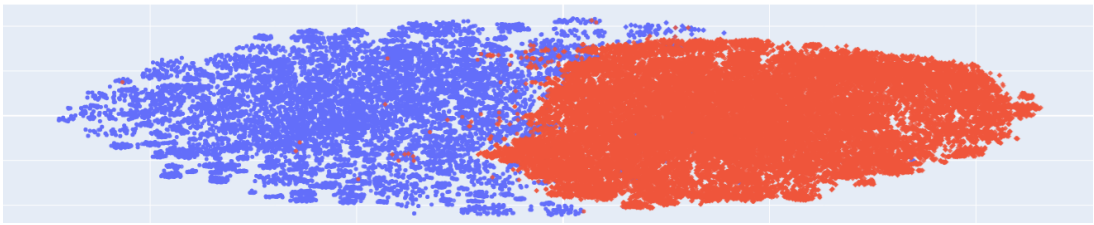

Figure 1: Embedding visualization showing separation between general-domain (blue, SQuADv2) and biomedical (red, PubMedQA) samples, with overlap pockets reflecting cross-domain similarity

mantic proximity while preserving broader category distinctions.

3.2. Data Augmentation

Prior work has shown that large models can be powerful engines for creating diverse training corpora (Puri et al., 2020; Wang et al., 2023; Mitra et al., 2024). While many black-box augmentation pipelines have proven effective in practice, our goal is more focused within a linguistic scope, aiming to maximize the contextual knowledge and generative ability of LLMs to produce questions or question–answer pairs from multiple linguistic angles. To isolate this effect, we employ a single model, `meta-llama/Llama-3.1-70B-Instruct`,[1] and keep the prompting setup consistent for both generation and inference. This design ensures that observed differences stem from the type of linguistic augmentation rather than engineering optimizations. In this way, we do not attempt to beat state-of-the-art accuracy, but rather to analyze and better understand the role of linguistic augmentation itself. The exact prompts used for each augmentation type are provided in Appendix A. SQuADv2 contains unanswerable items, which we retain in the augmentation pool. Lexical and syntactic edits preserve the unanswerable status, while semantic augmentation generates new questions grounded in the passage, marking them as unanswerable when no valid span exists. This concern does not apply to PubMedQA, which has no unanswerable category.

Within the $1/32$ supervision split, each original instance is expanded by calling the model twice per augmentation type (lexical, syntactic, semantic), producing two variants per category. Thus, one original question yields six synthetic counterparts. We then apply GPT-4o to validate whether the generated examples respect the intended category constraints.[2] After validation, we retain only the valid examples for fine-tuning. Table 2 summarizes the distribution of validation outcomes across datasets.

Label	SQuADv2	PubMedQA
Valid	17,768	19,050
Unsure	4	17
Invalid	7,207	5,927

Table 2: Validation outcomes for synthetic QA pairs across datasets. Only the *valid* examples are used for downstream fine-tuning

3.2.1. Lexical-based data augmentation

The question is rephrased by replacing words with synonyms, near-synonyms, or idiomatic expressions. The answer remains unchanged, but the surface form of the question differs.

- Original question: *At what age did Beyoncé meet LaTavia Robertson?* (SQuADv2)
- Lexical variant: *At what age did Beyoncé encounter LaTavia Robertson?*

3.2.2. Syntactic-based data augmentation

This strategy rewrites questions using alternate grammatical structures such as switching between active and passive voice or simple and complex forms while preserving both meaning and the original answer.

- Original question: same as above
- Syntactic variant: *LaTavia Robertson first met Beyoncé when she was how old?*

3.2.3. Semantic-based data augmentation

Unlike the above, semantic augmentation produces new question-answer that probe different aspects of the same passage. These are not strict paraphrases: answers may differ, yet they remain grounded in the same evidence.

[1] https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct

[2] We intentionally oversample at this stage, since a portion of synthetic examples are filtered out; we need to ensures that the final supervision proportions remain consistent across categories.

- Original context: *At age eight, Beyoncé and childhood friend Kelly Rowland met LaTavia Roberson [...] They were placed into a group with three other girls as Girl's Tyme, and rapped and danced on the talent show circuit in Houston. After seeing the group, R&B producer Arne Frager brought them to his Northern California studio and placed them in Star Search, the largest talent show on national TV at the time. Girl's Tyme failed to win [...]*
- Semantic variant: *What was the name of the talent show that Girl's Tyme failed to win?*
- Answer: *Star Search*

3.3. Models and Fine Tuning

We fine-tune two open-source instruction-following LLMs: `Llama-3.1-8B-Instruct` (Llama Community License) and `Qwen2.5-7B-Instruct` (Apache 2.0), abbreviated hereafter as Llama8B-I and Qwen7B-I respectively (with Llama70B-I denoting Meta-Llama-3.1-70B-Instruct used for augmentation). Choosing open-source backbones ensures reproducibility and alignment with community-driven policies. Fine-tuning uses LoRA (Hu et al., 2022) with rank r=16, α=16, and dropout 0.05, applied to all attention and feed-forward projection matrices, updating only 0.53% of parameters. We use AdamW with a learning rate of 1×10^{-4}, cosine schedule, warmup ratio 0.1, weight decay 0.01, and a per-device batch size of 32 with 4 gradient accumulation steps. Training runs for 4 epochs in bf16 precision. Full hyperparameters are listed in Appendix B. To set the training schedule, we experimented on the $\frac{1}{128}$ split with three seeds: evaluation loss diverged after epoch 4 and extending to 8 epochs yielded no gains, so we fix training to 4 epochs. Early trials also showed that generating multiple QA pairs per call caused hallucinations (Nayab et al., 2025); switching to one pair per call improved quality.

3.4. Experimental Workflow

We construct augmented training sets by mixing the original data with different types of linguistic edits *lexical*, *syntactic*, *semantic*, and a combined *all* variant while keeping the overall sample size fixed across conditions. After generating augmented data, we perform a quality-control step: each candidate QA pair is validated by `GPT-4o` (Hurst et al., 2024) acting as an LLM judge (Gu et al., 2025), which assesses whether the augmented items remain faithful to their supporting context and relevant to their edit types. In Section 5, we further compare the alignment of `GPT-4o` judgments with human annotations. From this filtered subset, we construct the final training data used for fine-tuning. In total, our setup covers:

- 2 datasets: SQuADv2 (general) and PubMedQA (biomedical)
- 2 models: Llama8B-I and Qwen7B-I
- 3 scales: downsampled subsets at $\frac{1}{128}$, $\frac{1}{64}$, and $\frac{1}{32}$ of the original training set (we omit $\frac{1}{16}$ due to time constraints, but fine-tune the baseline at this scale to enable comparison with lower-proportion counterparts)
- 5 augmentation conditions: baseline (no augmentation, purely train set), syntactic, lexical, semantic, and all (uniform mixture of the other categories)

This factorial design yields $2 \times 2 \times 3 \times 5 = 60$ runs overall. Together, these experiments allow us to systematically analyze how augmentation type, data scale, and model choice interact under low-resource conditions.

4. Evaluation

We assess model outputs with three metrics: *Exact Match (EM)*, *word-level F1*, and *semantic similarity* computed from sentence embeddings. Together, these capture strict string agreement, partial overlap, and meaning-level alignment. For PubMedQA, we substitute a simpler EM (*EM bin.*), which evaluates whether the predicted label (`yes`, `no`, or `maybe`) matches the gold annotation. At the same time, we retain word-level F1 and semantic similarity to evaluate the longer free-form reasoning answers, since they capture partial overlap and meaning beyond the discrete label. We do not compute span-level EM on PubMedQA, as answers are long sentences that rarely match exactly even GPT-4o achieves zero EM in this setting.

Normalization and multiple references. Following standard practice for contextual QA, predictions and references are normalized by lower-casing, removing punctuation and articles, and collapsing extra whitespace.

SQuADv2 Analysis. Our evaluation on SQuADv2, summarized in Table 3, reveals several key insights. First, supervised fine-tuning provides a performance boost over the vanilla models (i.e., base models used without any task-specific fine-tuning). For instance, the Qwen7B-I baseline, when fine-tuned on just the $1/128$ data subset, achieves 0.68 EM and 0.77 F1, outperforming its vanilla counterpart's 0.54 EM and 0.65 F1. This highlights the critical value of task-specific adaptation, even with minimal data. When comparing models, the fine-tuned

Model	EM	F1	Sim ↑
SQuADv2 (vanilla, i.e., no fine-tuning)			
Qwen7B-I	0.54	0.65	0.73 ± 0.38
Llama8B-I	0.57	0.67	0.74 ± 0.39
Llama70B-I	0.54	0.66	0.74 ± 0.38
GPT-4o (vanilla)	0.56	0.68	0.78 ± 0.33
SQuADv2 (1/128 subset)			
Qwen7B-I Baseline	0.68	0.77	0.83 ± 0.33
Qwen7B-I Semantic	0.69	0.77	0.83 ± 0.33
Qwen7B-I Syntactic	0.68	0.77	0.83 ± 0.33
Qwen7B-I Lexical	0.68	0.77	0.83 ± 0.33
Qwen7B-I All	0.68	0.77	0.83 ± 0.33
Llama8B-I Baseline	0.63	0.72	0.78 ± 0.37
Llama8B-I Semantic	0.60	0.70	0.76 ± 0.39
Llama8B-I Syntactic	0.62	0.71	0.77 ± 0.38
Llama8B-I Lexical	0.62	0.72	0.78 ± 0.38
Llama8B-I All	0.62	0.71	0.77 ± 0.38
SQuADv2 (1/64 subset)			
Qwen7B-I Baseline	0.72	0.79	0.84 ± 0.33
Qwen7B-I Semantic	0.68	0.76	0.82 ± 0.35
Qwen7B-I Syntactic	0.72	0.79	0.84 ± 0.33
Qwen7B-I Lexical	0.71	0.79	0.84 ± 0.33
Qwen7B-I All	0.71	0.79	0.84 ± 0.33
Llama8B-I Baseline	0.67	0.73	0.77 ± 0.40
Llama8B-I Semantic	0.60	0.69	0.75 ± 0.40
Llama8B-I Syntactic	0.65	0.73	0.77 ± 0.39
Llama8B-I Lexical	0.66	0.73	0.77 ± 0.40
Llama8B-I All	0.65	0.73	0.77 ± 0.39
SQuADv2 (1/32 subset)			
Qwen7B-I Baseline	0.57	0.67	0.74 ± 0.39
Qwen7B-I Semantic	0.55	0.66	0.76 ± 0.37
Qwen7B-I Syntactic	0.50	0.62	0.70 ± 0.40
Qwen7B-I Lexical	0.49	0.59	0.68 ± 0.42
Qwen7B-I All	0.52	0.63	0.71 ± 0.40
Llama8B-I Baseline	0.58	0.61	0.63 ± 0.47
Llama8B-I Semantic	0.55	0.64	0.70 ± 0.43
Llama8B-I Syntactic	0.57	0.60	0.64 ± 0.46
Llama8B-I Lexical	0.56	0.60	0.63 ± 0.46
Llama8B-I All	0.60	0.65	0.69 ± 0.44
SQuADv2 (1/16 subset)			
Qwen7B-I Baseline	0.73	0.80	0.85 ± 0.33
Llama8B-I Baseline	0.70	0.77	0.80 ± 0.36

Table 3: Evaluation results on SQuAD v2 subsets with different augmentation strategies

Qwen7B-I consistently outperforms Llama8B-I across most baseline settings, suggesting stronger data efficiency for Qwen on this benchmark. Surprisingly, our data augmentation strategies (Lexical, Syntactic, Semantic, All) offer no substantial benefit. Most of the variants perform identically to the baseline. This indicates that augmentation does not substantially alter performance. Notably, augmented models match the baseline despite seeing fewer unique source passages, suggesting that increased QA diversity compensates for reduced contextual breadth.

The relationship between data proportion and performance is notably non-monotonic. While results improve when scaling from the $1/128$ to the $1/64$ subset, they drop unexpectedly at $1/32$ before recovering at $1/16$. We hypothesize that this dip may reflect an interaction between dataset size and the dynamics of LoRA fine-tuning. With very small subsets (e.g., $1/128$), the model remains underfit but relatively stable; with larger subsets (e.g., $1/16$), the adapter has enough signal to generalize effectively. At intermediate scales, however, the model may cross a threshold where it overfits to limited but noisy supervision, leading to degraded performance. Such non-monotonic scaling behavior has been observed in other settings (Nakkiran et al., 2021), suggesting that careful calibration of data size and fine-tuning strategy is essential in low-resource environment.

Summary. For Qwen7B-I, augmentation yields parity with the baseline across splits while using fewer unique source passages suggesting that added QA diversity can substitute for contextual breadth (no single strategy dominates).

PubMedQA Analysis. Our results on PubMedQA (Table 4) differ markedly from SQuADv2. A key distinction is that PubMedQA's evaluation includes EM (bin.) that only checks whether the model predicted the correct label (*yes*, *no*, or *maybe*) and the rest reflect how well the model reasons in free-text explanations. This separation reveals an important phenomenon, some models are "stubborn" producing an answer reasoning without committing to the correct discrete label, which lowers EM but leaves F1 and similarity scores intact. For instance, Qwen7B-I at the $1/128$ subset achieves only 0.75 EM but 0.53 similarity, while its semantic-augmented variant keeps EM stable but lifts similarity to 0.75. This suggests that label prediction and reasoning fluency are not always aligned, and both need to be considered together.

When comparing augmentation styles and data proportions, several signals emerge. Semantic augmentation proves most helpful at the smallest scale, for Qwen7B-I, it improves F1 from 0.18 to 0.28 at $1/128$ and raises similarity by more than 0.20. In contrast, syntactic and lexical variants are less reliable, occasionally dropping EM sharply (e.g., Qwen7B-I Syntactic at $1/32$ falls to 0.42 EM) despite stable reasoning metrics. The "All" strategy produces middling results, rarely surpassing semantic alone. Scaling with more supervision does not yield monotonic improvements, performance rises from $1/128$ to $1/64$, then becomes erratic at $1/32$, and collapses at $1/16$. This instability suggests an interaction between noisy biomedical su-

Model	EM (bin.)	F1	Sim ↑
PubMedQA (vanilla)			
Qwen7B-I	0.84	0.27	0.75 ± 0.09
Llama8B-I	0.88	0.26	0.73 ± 0.14
Llama70B-I	0.85	0.26	0.7 ± 0.22
GPT-4o (vanilla)	0.76	0.26	0.70 ± 0.23
PubMedQA (1/128 subset)			
Qwen7B-I Baseline	0.75	0.18	0.53 ± 0.35
Qwen7B-I Semantic	0.76	0.28	0.75 ± 0.14
Qwen7B-I Syntactic	0.76	0.19	0.56 ± 0.33
Qwen7B-I Lexical	0.72	0.19	0.54 ± 0.34
Qwen7B-I All	0.73	0.19	0.54 ± 0.34
Llama8B-I Baseline	0.89	0.18	0.49 ± 0.37
Llama8B-I Semantic	0.89	0.19	0.54 ± 0.34
Llama8B-I Syntactic	0.88	0.20	0.53 ± 0.36
Llama8B-I Lexical	0.88	0.19	0.53 ± 0.36
Llama8B-I All	0.89	0.14	0.38 ± 0.38
PubMedQA (1/64 subset)			
Qwen7B-I Baseline	0.71	0.31	0.75 ± 0.19
Qwen7B-I Semantic	0.86	0.24	0.68 ± 0.24
Qwen7B-I Syntactic	0.73	0.31	0.75 ± 0.20
Qwen7B-I Lexical	0.70	0.28	0.71 ± 0.26
Qwen7B-I All	0.82	0.24	0.64 ± 0.30
Llama8B-I Baseline	0.72	0.33	0.80 ± 0.09
Llama8B-I Semantic	0.88	0.18	0.52 ± 0.35
Llama8B-I Syntactic	0.71	0.33	0.80 ± 0.10
Llama8B-I Lexical	0.66	0.33	0.80 ± 0.09
Llama8B-I All	0.86	0.30	0.77 ± 0.13
PubMedQA (1/32 subset)			
Qwen7B-I Baseline	0.87	0.33	0.80 ± 0.09
Qwen7B-I Semantic	0.89	0.25	0.73 ± 0.12
Qwen7B-I Syntactic	0.42	0.33	0.81 ± 0.09
Qwen7B-I Lexical	0.56	0.33	0.79 ± 0.12
Qwen7B-I All	0.70	0.31	0.77 ± 0.16
Llama8B-I Baseline	0.38	0.33	0.80 ± 0.09
Llama8B-I Semantic	0.88	0.15	0.54 ± 0.26
Llama8B-I Syntactic	0.39	0.33	0.80 ± 0.09
Llama8B-I Lexical	0.39	0.33	0.80 ± 0.09
Llama8B-I All	0.82	0.33	0.80 ± 0.10
PubMedQA (1/16 subset)			
Qwen7B-I Baseline	0.36	0.33	0.8 ± 0.09
Llama8B-I Baseline	0.38	0.33	0.8 ± 0.09

Table 4: Evaluation results on PubMedQA subsets with different augmentation strategies

pervision and LoRA adaptation.

Model comparison reinforces these findings: Llama8B-I achieves strong EM at $1/128$ (0.89) but drops to 0.38 at $1/32$, while Qwen7B-I adapts better with more supervision, likely reflecting differences in tokenizer coverage and pretraining bias. Augmentation thus interacts with both model type and supervision size.

Summary. For PubMedQA, semantic augmentation is the only strategy that shows consistent gains, improving Qwen7B-I at the $1/128$ scale (higher F1 and similarity) without hurting EM. Syntactic and lexical variants are unstable, especially at $1/32$, while the mixed "All" setting rarely outperforms semantic alone. Llama8B-I performs best at extreme low-data ($1/128$), but Qwen7B-I adapts better at moderate scale ($1/32$).

5. Human Validation

Large language models can generate fluent but unfaithful content (Huang et al., 2025; Kalai et al., 2025), so we manually validate a subset of the synthetic QA data to quantify quality and calibrate an automatic screener. Our annotation pool comprises one Linguist, one PhD student in NLP, and two NLP Experts. For each dataset, we sample 75 items and assign one of three labels: *invalid*, *unsure*, and *valid*. The labeling guidelines differ slightly across augmentation types to reflect their specific failure modes (e.g., lexical vs. semantic drift). In parallel, we ask GPT-4o to label the same items using the exact guideline, in order to test whether it can reliably scale the validation to the full corpus.

5.1. Validation Protocol and Confidence Estimation

Protocol and timing. Annotators worked independently with the same instruction sheet and examples. After annotation, we measure the level of consensus between human raters and the GPT-4o outputs to assess whether the model can approximate human judgment. Manual validation is costly: 36–67 minutes per 75 SQuADv2 items and 1.5–3 hours for PubMedQA (domain-specific terms require slower reading). In contrast, GPT-4o labels the same batch in about 250 s (SQuADv2) and 370 s (PubMed).

Confidence Estimation from Log-Probabilities. To avoid repeated API calls for calibration, we approximate model confidence directly from the log-probability of the emitted label token ("-1", "0", or "1" referring to invalid, unsure, and valid labels). Specifically, we extract the log-prob reported for that token and convert it into a probability by exponentiation. This yields a single confidence score per item without additional queries. While approximate, this proxy reflects the model's internal preference for the chosen label and is commonly used in lightweight uncertainty estimation for LMs (Kadavath et al., 2022).

Figure 2 plots the distribution of GPT-4o confidence grouped by the human reference label. The model is highly confident overall, with mass near 1.0 for both *valid* and *invalid* cases. The valid class shows slightly broader spread (some outliers < 0.9), whereas invalid items are predicted with near-perfect certainty. Interestingly, the model never outputs the intermediate *unsure* label (as in our large-scale experiments, which yielded only a

negligible number of such cases) and thus shows no explicit uncertainty. This pattern indicates *overconfidence*: the model differentiates valid from invalid on average, but leaves little room for calibrated uncertainty, which cautions against fully automatic acceptance without human.

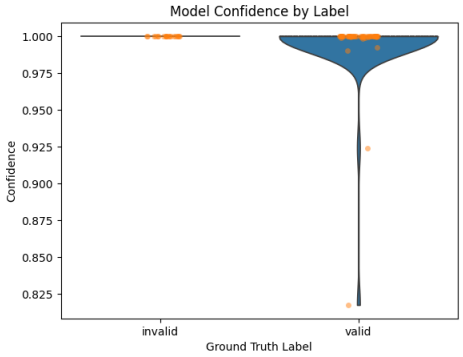

Figure 2: Model confidence (from label log-probs). Both datasets show the same pattern; we plot one dataset (SQuADv2) for brevity

5.2. Error Taxonomy from Human Review

Beyond a single validity label, annotators consistently flagged recurrent issues in synthetic QA. We group them into five categories:

- Grammar & fluency (bad grammatical phrasing)
- Faithfulness to context (missing key details or unsupported additions)
- Answer–question alignment (answer is over/under-specified relative to the question)
- Redundancy/minimal variation (near-duplicates or trivial rewrites)
- Clarity & specificity (vague wording, ambiguous acronyms, underspecified entities)

This taxonomy offers a structured lens for diagnosing weak points in synthetic QA. Going forward, these categories could be used not only for evaluation but also to guide augmentation itself. For instance, by adapting prompts to reduce redundancy, enforce grounding for faithfulness, or encourage clearer wording. They could also serve as automatic filter signals, training lightweight detectors that flag likely errors before human review.

5.3. Human vs. Model Validation

An important question is how closely model-based validation aligns with human experts, and whether this alignment shifts across domains. Manual review is slow and costly, whereas models like GPT-4o are efficient but may overlook domain subtleties. We therefore compare validity rates, annotator confidence, and category-level differences between humans and the model.

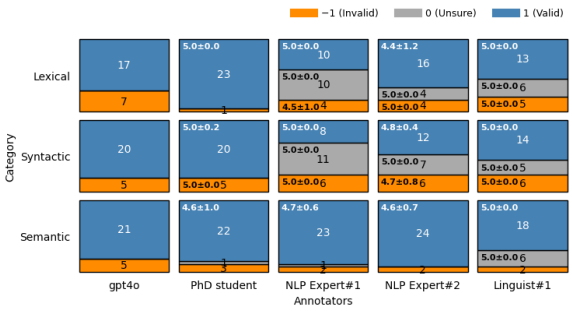

Figure 3: SQuADv2 validation (75 QA pairs). Agreement is relatively high, with disagreements concentrated in syntactic vs. semantic categories

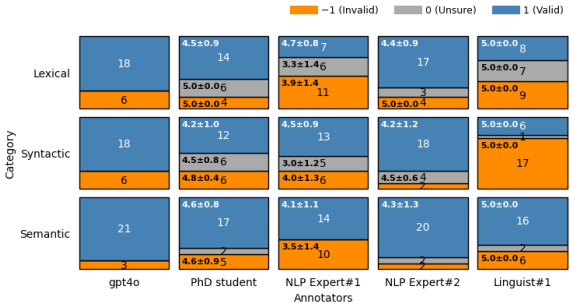

Figure 4: PubMedQA validation (75 QA pairs). NLP Experts marked more items as *invalid* or *unsure*, highlighting domain sensitivity absent in the model

The figures 3 and 4 present both the human evaluation (four annotators) and the automatic evaluation (using GPT-4o) for the three kinds of data augmentation (lexical, syntactic, and semantic) on the SQuADv2 and PubMedQA datasets (75 QA pairs evaluated on each dataset). We represent in blue the amount of valid generated data, in orange the amount of invalid generated data, and in grey the amount of generated data for which uncertainty was considered. Since human evaluators could assign a confidence score, the standard deviation is also indicated in each box.

On SQuADv2 (figure 3), humans judged 81.5% of items valid versus 77.3% for GPT-4o, a small gap of Δ Valid% $= -4.2$ (negative = model stricter). At the augmentation level, GPT-4o was more lenient on syntactic data $(+9.9)$ but stricter on semantic (-9.9) and lexical (-10.7). Overall, the model broadly tracked human judgments with mild shifts by augmentation type. Confidence remained high (4.8 ± 0.4), suggesting disagreements stem from interpretation rather than uncertainty.

For PubMedQA (figure 4), the pattern reverses, humans judged 66.4% valid, GPT-4o 79.2% ($\Delta = +12.8$). This permissive bias persisted across semantic $(+13.1)$, syntactic $(+13.7)$, and lexical $(+12.8)$. NLP Experts applied stricter criteria, introducing more *invalid* and *unsure* labels not captured

by the model. Confidence dropped to 4.5 ± 0.7, reflecting the difficulty of biomedical texts, where domain-specific terminology and higher stakes reduce annotator certainty (Wang et al., 2021).[3]

Taken together, results reveal a domain-dependent shift. In general data (SQuADv2), humans and GPT-4o converge, with disagreements mostly stylistic. In specialized domains (PubMed), humans adopt stricter thresholds while the model remains permissive, widening the gap. A practical implication is that GPT-4o serves well as a fast first-pass screener, leaving experts to review harder edge cases where consensus is fragile.

6. Energy and Emission Analysis

As tracking carbon is increasingly important both to make the environmental costs of ML visible, recent work urges routine reporting of energy and emissions for ML experiments (Strubell et al., 2019; Schwartz et al., 2020). We track energy use and emissions with CodeCarbon[4] (Courty et al., 2024), an open-source Python library that samples hardware power draw during runtime and converts it into total energy (kWh) equivalent emissions. All fine-tuning experiments were run on a single node equipped with 2× NVIDIA H100 PCIe GPUs and an Intel® Xeon® Gold 5418Y CPU.

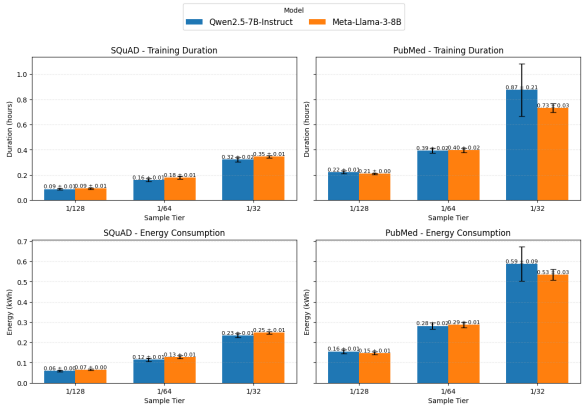

Figure 5: Training time (h) and energy use (kWh) across datasets, sampling tiers, and models. Bars show mean over augmentation types; error bars indicate the standard error of the mean (SEM)

Figure 5 reports training duration (top) and energy consumption (bottom) for SQuADv2 and PubMedQA at three sampling tiers ($\frac{1}{128}$, $\frac{1}{64}$, $\frac{1}{32}$). Results are averaged over the five augmentation conditions (Baseline, Lexical, Syntactic, Semantic, All). Error bars denote standard error, but differences are minimal, indicating that augmentation choice has negligible impact on runtime or energy.

On SQuADv2, Qwen7B-I is consistently faster and slightly more energy-efficient than Llama8B-I, reflecting its smaller parameter count. In contrast, on PubMedQA the trend reverses, with Qwen taking longer and consuming more energy. We hypothesize this stems from differences in tokenization and domain vocabulary. PubMedQA's biomedical terms may yield longer effective sequences for Qwen. This highlights how efficiency can depend not only on parameter count but also on model–data interaction (Hoffmann et al., 2022; Zhou et al., 2024).

7. Conclusion

We systematically compared lexical, syntactic, and semantic data augmentation for extractive QA under low-resource conditions, with fidelity-aware validation via GPT-4o and human audit. Our experiments across SQuADv2 (general) and PubMedQA (biomedical) yield three main findings: (i) semantic augmentation is the only strategy that consistently improves biomedical QA at the smallest scale ($1/128$), while carefully curated baselines often suffice in the general domain; (ii) performance scales non-monotonically with data size, revealing an interaction between LoRA adaptation and supervision volume that warrants careful calibration; and (iii) label prediction can diverge from reasoning quality, underscoring the need for multi-metric evaluation beyond headline accuracy.

Several limitations qualify these conclusions. We evaluated only two model families (Qwen7B-I and Llama8B-I); other architectures such as Mistral-7B may respond differently to the same augmentation types (Liu et al., 2024). Our domain coverage is limited to one general and one biomedical benchmark, legal (Guha et al., 2023) and financial QA (Chen et al., 2021) pose structurally different challenges that future work should address. The GPT-4o validation loop, while efficient, was not calibrated against human preferences, prompt tuning or preference alignment of the judge could further improve augmentation filtering. Finally, we did not scale beyond $1/16$ of the original training sets; larger fractions may reveal additional nonlinearities due to optimizer dynamics or augmentation quality variance. Despite these limitations, we believe this work provides practical guidelines for choosing augmentation strategies under data scarcity and opens the door to more principled, quality-focused augmentation pipelines for domain-adaptable QA.

[3] Confidence values (1–5 scale) are shown inside the bars. For counts below five, values are omitted; in most cases, the score was 5.

[4] https://pypi.org/project/codecarbon/

8. Ethical Considerations

We rely only on public datasets (SQuADv2, PubMedQA) under their respective licenses and do not process personally identifiable information (PII) or protected health information (PHI). All synthetic data are clearly labeled and generated with open models (Llama70B-I for augmentation; Llama8B-I and Qwen7B-I for fine-tuning) to ensure transparency and reproducibility. Human validation was carried out by a small team of expert annotators who participated voluntarily; no demographic or sensitive data were collected. To account for environmental impact, we tracked hardware, training time, and energy consumption using CodeCarbon. Whenever possible, we favored shorter training schedules and smaller models to reduce resource usage while maintaining accuracy.

Code is available at https://anonymous.4open.science/r/ssl-4CB8.

9. 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.

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott McCarley, Michael McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej Sakrajda, Avi Sil, Rosario Uceda-Sosa, Todd Ward, and Rong Zhang. 2020. The TechQA dataset. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1269–1278, Online. Association for Computational Linguistics.

Yung-Chieh Chan, George Pu, Apaar Shanker, Parth Suresh, Penn Jenks, John Heyer, and Sam Denton. 2024. Balancing cost and effectiveness of synthetic data generation strategies for llms. *arXiv preprint arXiv:2409.19759*.

Jie Chen, Yupeng Zhang, Bingning Wang, Wayne Xin Zhao, Ji-Rong Wen, and Weipeng Chen. 2024. Unveiling the flaws: Exploring imperfections in synthetic data and mitigation strategies for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14855–14865, Miami, Florida, USA. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Arijit Chowdhury and Aman Chadha. 2024. Generative data augmentation using llms improves distributional robustness in question answering. In *Proceedings of the 18th conference of the European chapter of the association for computational linguistics: student research workshop*, pages 258–265.

Benoit Courty, Victor Schmidt, Sasha Luccioni, Goyal-Kamal, MarionCoutarel, Boris Feld, Jérémy Lecourt, LiamConnell, Amine Saboni, Inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, Marc Alencon, Michał Stęchły, Christian Bauer, Lucas Otávio N. de Araújo, JPW, and MinervaBooks. 2024. mlco2/codecarbon: v2.4.1.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Fang Zeng, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2025. Auggpt: Leveraging chatgpt for text data augmentation. *IEEE Transactions on Big Data*, 11(3):907–918.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Dan Friedman and Adji Bousso Dieng. 2023. The vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A survey on LLM-as-a-Judge.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. Legalbench: a collaboratively built benchmark for measuring legal reasoning in large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Zhen Guo, Peiqi Wang, Yanwei Wang, and Shangdi Yu. 2023. Improving small language models on pubmedqa via generative data augmentation. In *LLM4AI'23: Workshop on Foundations and Applications in Large-scale AI Models -Pre-training, Fine-tuning, and Prompt-based Learning*, Long Beach, CA, USA.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. ArXiv preprint arXiv:2203.15556.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. Technical report, OpenAI.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. 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 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Qiao Jin, Zheng Yuan, Guangzhi Xiong, Qianlan Yu, Huaiyuan Ying, Chuanqi Tan, Mosha Chen, Songfang Huang, Xiaozhong Liu, and Sheng Yu. 2022. Biomedical question answering: A survey of approaches and challenges. *ACM Comput. Surv.*, 55(2).

Saurav Kadavath, Tom Conerly, Amanda Askell, T. J. Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zachary Dodds, Nova Dassarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, John Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom B. Brown, Jack Clark, Nicholas Joseph, Benjamin Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *ArXiv*, abs/2207.05221.

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2025. Why language models hallucinate. ArXiv preprint 2509.04664v1.

Julien Khlaut, Corentin Dancette, Elodie Ferreres, Alaedine Bennani, Paul Hérent, and Pierre Manceron. 2024. Efficient medical question answering with knowledge-augmented question generation. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 10–20, Mexico City, Mexico. Association for Computational Linguistics.

Qin Liu, Fei Wang, Nan Xu, Tianyi Lorena Yan, Tao Meng, and Muhao Chen. 2024. Monotonic paraphrasing improves generalization of language model prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9861–9877, Miami, Florida, USA. Association for Computational Linguistics.

Shayne Longpre, Yi Lu, Zhucheng Tu, and Chris DuBois. 2019. An exploration of data augmentation and sampling techniques for domain-agnostic question answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 220–227, Hong Kong, China. Association for Computational Linguistics.

Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweti Mahajan, Dany Rouhana, Andres Codas, Yadong Lu, Wei ge Chen, Olga Vrousgos, Corby Rosset, Fillipe Silva, Hamed Khanpour, Yash

Lara, and Ahmed Awadallah. 2024. Agentinstruct: Toward generative teaching with agentic flows.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. 2021. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003.

Sania Nayab, Giulio Rossolini, Marco Simoni, Andrea Saracino, Giorgio Buttazzo, Nicolamaria Manes, and Fabrizio Giacomelli. 2025. Concise thoughts: Impact of output length on LLM reasoning and cost. ArXiv preprint arXiv:2407.19825.

Sergey Pletenev, Maria Marina, Daniil Moskovskiy, Vasily Konovalov, Pavel Braslavski, Alexander Panchenko, and Mikhail Salnikov. 2025. How much knowledge can you pack into a lora adapter without harming llm? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4309–4322.

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, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

Revanth Gangi Reddy, Bhavani Iyer, Md Arafat Sultan, Rong Zhang, Avi Sil, Vittorio Castelli, Radu Florian, and Salim Roukos. 2020. End-to-end QA on COVID-19: Domain adaptation with synthetic training. ArXiv preprint arXiv:2012.01414.

Maximilian Schmidt, Andrea Bartezzaghi, and Ngoc Thang Vu. 2024. Prompting-based synthetic data generation for few-shot question answering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13168–13178.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green ai. *Commun. ACM*, 63(12):54–63.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Minju Seo, Jinheon Baek, James Thorne, and Sung Ju Hwang. 2024. Retrieval-augmented data augmentation for low-resource domain tasks. ArXiv preprint arXiv:2402.13482v1.

Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-end synthetic data generation for domain adaptation of question answering systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460, Online. Association for Computational Linguistics.

Marco Antonio Sobrevilla Cabezudo, Marcio Lima Inacio, and Thiago Alexandre Salgueiro Pardo. 2024. Investigating paraphrase generation as a data augmentation strategy for low-resource AMR-to-text generation. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 663–675, Tokyo, Japan. Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Guy Tevet and Jonathan Berant. 2021. Evaluating the evaluation of diversity in natural language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 326–346, Online. Association for Computational Linguistics.

Kanix Wang, Robert Stevens, Halima Alachram, Yu Li, Larisa Soldatova, Ross King, Sophia Ananiadou, Maolin Li, Fenia Christopoulou, Jose Luis Ambite, Joel Matthew, Sahil Garg, Ulf Hermjakob, Daniel Marcu, Emily Sheng, Tim Beißbarth, Edgar Wingender, Aram Galstyan, and Andrey Rzhetsky. 2021. Nero: a biomedical named-entity (recognition) ontology with a large, annotated corpus reveals meaningful associations

through text embedding. *npj Systems Biology and Applications*, 7.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Joonho Yang, Seunghyun Yoon, Hwan Chang, Byeongjeong Kim, and Hwanhee Lee. 2025. Hallucinate at the last in long response generation: A case study on long document summarization. ArXiv preprint arXiv:2505.15291v2.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *Proceedings of ICLR*, Vancouver, Canada.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, Shengen Yan, Guohao Dai, Xiao-Ping Zhang, Yuhan Dong, and Yu Wang. 2024. A survey on efficient inference for large language models. ArXiv preprint arXiv:2404.14294v3.

Yuchang Zhu, Huizhe Zhang, Bingzhe Wu, Jintang Li, Zibin Zheng, Peilin Zhao, Liang Chen, and Yatao Bian. 2025. Measuring diversity in synthetic datasets. In *Forty-second International Conference on Machine Learning*.

A. Augmentation Prompts

Below are the prompts supplied to `Llama-3.1-70B-Instruct` for each augmentation type. Each prompt receives the original context, question, and answer as input, with temperature 0.6 and top-p 0.9.

Lexical augmentation prompt.

```
You are given a context and an
original question-answer pair.
Rephrase the question using
different vocabulary and word
choices while maintaining the
same meaning and answer.

Return the result strictly in
JSON with the following struc-
ture: {"questions": ["..."],
"answers": ["..."]}
```

Syntactic augmentation prompt.

```
You are given a context and an
original question-answer pair.
Rewrite the question using dif-
ferent grammatical structure
or sentence construction while
preserving the same meaning and
answer.

Return the result strictly in
JSON with the following struc-
ture: {"questions": ["..."],
"answers": ["..."]}
```

Semantic augmentation prompt.

```
You are given a context and an
original question-answer pair.
Create exactly 1 new question-
short answer pair that asks about
a different facet of the context
(not a paraphrase). You need to
be creative.

Return the result strictly in
JSON with the following struc-
ture: {"questions": ["..."],
"answers": ["..."]}
```

B. Hyperparameters

Hyperparameter	Value
LoRA rank (r)	16
LoRA α	16
LoRA dropout	0.05
Target modules	q/k/v/o/gate/up/down_proj
Optimizer	AdamW (fused)
Learning rate	1×10^{-4}
LR scheduler	Cosine
Warmup ratio	0.1
Weight decay	0.01
Max gradient norm	0.8
Batch size (per device)	32
Gradient accumulation	4
Epochs	4
Max sequence length	2048
Precision	bf16 + tf32
Seed	42

Table 5: Full hyperparameters used for LoRA fine-tuning