# Synthetic Data Generation for Training Diversified Commonsense Reasoning Models

**Anonymous ACL submission**

## Abstract

Conversational agents are required to respond to their users not only with high quality (i.e. commonsense bearing) responses, but also considering multiple plausible alternative scenarios, reflecting the diversity in their responses. Despite the growing need to train diverse commonsense generators, the progress of this line of work has been significantly hindered by the lack of large-scale high-quality diverse commonsense training datasets. Due to the high annotation costs, existing Generative Commonsense Reasoning (GCR) datasets are created using a small number of human annotators, covering only a narrow set of commonsense scenarios. To address this training resource gap, we propose a two-stage method to create **CommonSyn**, the first-ever synthetic dataset for diversified GCR. Large Language Models (LLMs) fine-tuned on CommonSyn show simultaneous improvements in both generation diversity and quality compared with vanilla models and models fine-tuned on manually annotated datasets.[1]

## 1 Introduction

GCR requires models to produce diverse, high-quality sentences from limited concepts, relying on intrinsic or external knowledge (Lin et al., 2020; Liu et al., 2023). In GCR, a Natural Language Generation (NLG) model is expected to generate sentences that are both *quality-bearing* (i.e. logically coherent and commonsense-aware) and *diverse* (i.e. offering varied viewpoints) based on limited input information (i.e. few concepts) relying on the models' intrinsic or externally retrieved knowledge (Liu et al., 2023; Yu et al., 2022; Hwang et al., 2023). Although CommonGen (Lin et al., 2020) serves as a standard benchmark for GCR, it suffers from critical data limitations: (1) **limited diversity**, as 83% of concept sets have fewer than two references; and (2) **quality issues**, including near-duplicates and incomplete structures. Consequently, models fine-tuned on existing datasets often struggle to balance quality with semantic diversity or generalise to unseen concepts.

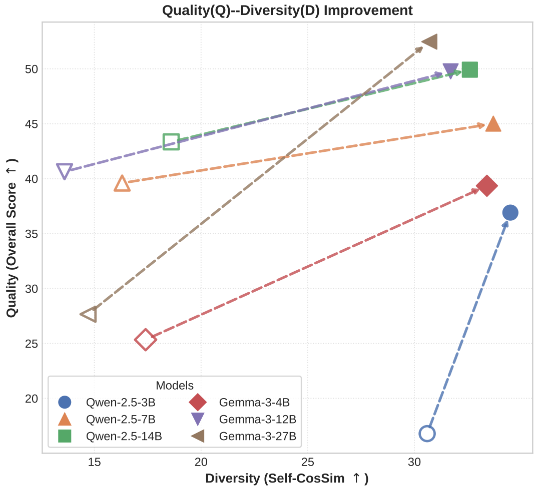

Figure 1: **Quality–Diversity trade-off for representative models.** The x-axis represents the semantic diversity (Self-CosSim, ↑) and the y-axis represents the generation quality (Overall, ↑). While vanilla models (hollow markers) often suffer from either low quality or limited diversity, its fine-tuned version on our synthetic data, CommonSyn (solid markers), consistently pushes the performance frontier towards the top-right quadrant, achieving a superior Pareto improvement across diverse model families.

Synthetic data generation offers a scalable solution (Yu et al., 2024; Guo et al., 2024). Compared to human annotations, it is easier to obtain scaleable and controllable quality data using LLMs (Bauer et al., 2024; Liu et al., 2024). However, recent studies have shown that synthetic data often face challenges such as low structural diversity, strong stylistic biases, and model collapse (Yu et al., 2023; Shumailov et al., 2023). Although

---

[1] The dataset and code are submitted to Open Review and will be publicly released upon paper acceptance.

recent work has attempted to diversify synthetic data (Yu et al., 2023; Miranda et al., 2023; Ge et al., 2024), they mainly optimise the diversity of the *data itself* rather than the Quality-Diversity (Q-D) trade-off of the fine-tuned model-output. It remains unclear how to create synthetic data for GCR so that both quality and diversity improve simultaneously after fine-tuning.

To bridge the gap, we propose **CommonSyn**, the first-ever synthetic dataset specifically designed for diversifying GCR considering CommonGen (Lin et al., 2020), a widely-studied GCR setup. We propose a two-stage method that evaluates multiple concept expansion and sample selection strategies to identify the optimal Q-D frontier. By balancing the local diversity within each concept set and the global diversity across the entire dataset, CommonSyn not only improves the quality but also the diversity of several LLMs when fine-tuned, compared to using the manually annotated CommonGen training set. Furthermore, those fine-tuned models further improve performance of several other GCR tasks, demonstrating the generalisability of CommonSyn.

Our main contributions are as follows:

- We propose **CommonSyn**, a synthetic dataset that enables fine-tuned models to consistently outperform those trained on human-annotated data in both quality and diversity.
- We propose a 2-stage data selection method that balances local diversity within each concept set as well as global diversity throughout the dataset, which is computationally efficient compared to gradient-based selection methods (Jung et al., 2025) that require computing loss gradients over model parameters.
- Extensive experiments across 11 LLMs demonstrate robust Pareto improvements. Furthermore, models fine-tuned on CommonSyn generalise well to related tasks such as abductive reasoning and story completion.

## 2 Related Work

**Diversifying the GCR.** To diversify GCR, a model must generate both commonsense-bearing and diverse sentences. Datasets such as CommonGen (Lin et al., 2020) provide a set of concepts and a set of sentences that describe various commonsense relations among those concepts, while ComVE (Wang et al., 2020) requires a GCR method to explain why a given counterfactual statement does not make commonsense. Methods for improving diversity in GCR have been explored from various angles. Sampling-based decoding strategies, such as nucleus sampling (Holtzman et al., 2019) improve diversity by sampling next tokens from the nucleus of the generation probability distribution. Other approaches integrate external information from a knowledge graph (Hwang et al., 2023) or retrieve sentences from large corpora (Liu et al., 2023; Cui et al., 2024) to increase the diversity of the generation. Zhang et al. (2024) used human-written prompts to explicitly instruct LLMs to generate diverse outputs. However, to the best of our knowledge, existing methods focus only on improving the diversity of model outputs, neither creating synthetic training data nor addressing the challenge of jointly improving both generation quality and diversity.

**Synthetic Data Creation.** Synthetic data refers to artificially generated data that mimics the characteristics of real-world data (Liu et al., 2024). LLMs have been used for the generation of synthetic data to augment questions or solutions (Jung et al., 2025; Yu et al., 2024). Previous studies used synthetic data for fine-tuning (Huang et al., 2025; Chen et al., 2024b), instruction tuning (Li et al., 2024; Wu et al., 2024), knowledge transfer (Meng et al., 2022) and evaluation (Zhu et al., 2025). To avoid the model collapse problem (i.e., generating highly similar data leading to diminishing model performance), methods have been proposed to diversify synthetic data (Feng et al., 2025). Ge et al. (2024) generated various prompts with customised fine-grained personas. Wang et al. (2025) augmented the training data by training a diversification model. Jung et al. (2025) generated data based on the gradient vector of a proxy model. Although methods for diversifying synthetic data have been proposed (Chen et al., 2024a; Zhu et al., 2025), they do not evaluate the joint Q-D of the fine-tuned model's outputs.

**Data Selection.** Data selection is the task of selecting a subset of desirable training samples from a larger training dataset, where automatic data selection methods have been proposed (Sener and Savarese, 2017; Kaushal et al., 2019; Xia et al., 2022). According to Qin et al. (2024), existing strategies can be categorised into three groups: (a) **Quality-based** methods that filter data using statistical indicators such as perplexity (Ankner et al., 2024) or LLM-based scoring (Chen et al., 2023) to ensure correctness, (b) **Diversity-based** methods that aim to reduce redundancy or improve coverage by clustering or sampling from the embedding space (Sener and Savarese, 2017; Tirumala et al., 2023; Xie et al., 2023), and (c) **Importance-based** methods identify influential samples that significantly impact model parameters using gradient information (Deng et al., 2024; Pan et al., 2024; Jung et al., 2025). **However, to the best of our knowledge, we are the first to create a synthetic dataset by combining those selection strategies to jointly improve quality and diversity in GCR.**

## 3 Synthetic Data Generation for GCR

### 3.1 Problem Setting

GCR tasks such as CommonGen (Lin et al., 2020) require a model to produce a natural, coherent, and plausible description of a scenario given a set of 3-5 concepts. For example, given the concept set "*bicycle*, *ride*, *street*", a plausible sentence could be "*The woman rides a bicycle down the street instead of walking.*" Formally, given a concept set $\mathcal{X} = \{x_1, \ldots, x_m\}$, the goal is to generate a sentence $y$ that covers all concepts in $\mathcal{X}$ while maintaining grammatical correctness and commonsense plausibility. Beyond producing a single sentence, the diversified GCR aims to generate a set of sentences $\mathcal{Y} = \{y_1, \ldots, y_n\}$, such that each individual sentence is *quality-bearing*, while the set of generated sentences collectively exhibits *diversity* (Li et al., 2016). Unlike decoding strategies that diversify output during inference (Holtzman et al., 2019), we focus on the **training data**. We construct a synthetic dataset named **CommonSyn**, $\mathcal{D}_{syn} = \{(\mathcal{X}_i, \mathcal{Y}_i)\}_{i=1}^{N}$, designed to improve the Pareto frontier of quality and diversity for models fine-tuned on it, resulting in a better quality-diversity trade-off on unseen test sets.

### 3.2 Concept Set Generation

Constructing high-quality concept sets for synthetic GCR is challenging as it must satisfy two conflicting criteria: (i) **Coherence:** concepts must be sufficiently semantically related to form a plausible daily scenario, and (ii) **Coverage:** the concept sets should cover a wide range of unique concepts and combinations to prevent a model from overfitting to a restricted body of knowledge. The original CommonGen concept sets are limited in scale and sentences per concept set (Table 1), but they provide an important structural insight: concepts appearing together typically lie within 2-hop distance in ConceptNet (Speer et al., 2017), indicating strong semantic and commonsense relations. Therefore, we use this property to design a simple yet effective concept expansion procedure.

| **Statistics** | **CommonSyn** | **CommonGen Train** |
|---|---|---|
| # Concept-Sets | 20, 742 | 32,651 |
| – Size = 3 | 9,847 | 25,020 |
| – Size = 4 | 6,857 | 4,240 |
| – Size = 5 | 4,038 | 3,391 |
| # Sentences | 83,184 | 67,389 |
| per Concept-Set | 4.01 | 2.06 |
| Average Length | 14.19 | 10.54 |

Table 1: Comparison between our synthetic dataset CommonSyn and the CommonGen training set.

Instead of using complete concept sets only from the human-annotated CommonGen training data (which limits diversity) or generating concepts from scratch (which often yields unrelated terms), we use the existing CommonGen training data as a candidate set of seeds. Specifically, given a concept set from the CommonGen training set, we randomly sample two concepts as *anchors* to instruct an LLM to generate 1-3 additional bridge concepts to form a novel, commonsense-bearing scenario (prompt details in Figure 4). This method ensures that the generated concepts maintain the semantic compatibility encoded in CommonGen while exploring novel combinations of concepts. For example, given the original concept set $\{$*bicycle*, *ride*, *street*$\}$, we randomly select *bicycle* and *ride* as anchors. In this case, LLM might generate *path* as an additional concept, resulting in a new concept set. This approach is based on the observation that in the original CommonGen (Lin et al., 2020) 50% of concept sets are fully connected within a distance of 2-hops, ensuring semantic coherence in the expanded concept set.

### 3.3 Sentence Generation

Given the expanded concept sets, we use three distinct prompting strategies to generate candidate sentences. Each strategy targets a different region in the Q-D space (see Table 3), producing $N$ candidates per concept set.[2] All outputs are limited to $\leq 22$ words to match the task distribution. The prompts used for this generation step are shown in Appendix E.

---

[2] We use $N = 4$, which is in agreement with the number of reference sentences included in CommonGen test sets.

- **Dynamic Few-Shot Generation** ($D_{dyn}$): We generate sentences one at a time. For each generation, we randomly sample $k = 5$ few-shot examples from the CommonGen training set. Although this follows the standard prompting protocol (Lin et al., 2020), we observe that despite the randomised examples, the LLM tends to generate highly similar sentences.

- **Multi-sentence Few-shot Generation** ($D_{ms}$): This strategy also uses few-shot prompting, but unlike $D_{dyn}$, we instruct the LLM to produce $N$ *different* candidate sentences within a single generation step (Zhang et al., 2024). By placing multiple outputs in the same context window, the model is implicitly encouraged to diversify subsequent sentences to avoid repetition. This strategy improves both lexical and syntactic variation, although the quality may fluctuate compared to the single step generation approach.

- **Chain-of-Thoughts (CoT)-guided Generation** ($D_{cot}$): CoT has significantly improved the reasoning abilities of LLMs in a wide range of tasks (Wei et al., 2022; Naik et al., 2023). To capture deeper semantic relations between concepts, we prompt the LLM to first produce a short explanation describing how the input concepts might relate to each other, and then require it to output the desired sentence one at a time. This reasoning step encourages the model to explicitly consider causal or temporal relations before composing a sentence, leading to different styles of candidate sentences compared to those generated with $D_{dyn}$ or $D_{ms}$.

After the generation step, we apply a **keyword coverage filter** to ensure all input concepts appear in the output sentences. We combine all remaining sentences into a unified candidate pool. As shown later in § 4.3, naïvely merging the candidates (12 sentences per concept set) from the three strategies does not improve the Q–D characteristics of the fine-tuned model, calling for a principled data selection approach.

### 3.4 Data Selection

To construct the final **CommonSyn** dataset, we use a two-stage method that balances *local* diversity (within a concept set) and *global* quality/coverage.

**Local Selection.** For each concept set $\mathcal{C}$, we first filter out low-quality candidates. We use a lightweight scorer (Gemini-2.5-flash) to assign a plausibility score $Q(y) \in [1, 10]$ (with 10 indicating the most plausible scenarios) to each sentence, scoring for its grammatical correctness and commonsense validity (Figure 6). Then we discard those with $Q(y) < 4$.

For the remaining candidates $\mathcal{Y_C}$ within the same concept set, we compute their average pairwise cosine similarity using SimCSE sentence embeddings (Gao et al., 2021). Specifically, for a sentence $y_i$, its local diversity score, $D_{\text{local}}(y_i)$, is defined as:

$$D_{\text{local}}(y_i) = 1 - \frac{1}{|\mathcal{Y_C}| - 1} \sum_{y_j \in \mathcal{Y_C}, j \neq i} \cos(\mathbf{e}_i, \mathbf{e}_j). \tag{1}$$

We then select the top–$k$ sentences with the highest $D_{\text{local}}$ scores for each concept set (setting $k = 8$ in our experiments) to maximise the **local diversity**. This step ensures that our synthetic dataset maintains high intra-concept-set variety.

**Global Selection.** We aggregate all locally selected candidates into a global pool $\mathcal{S}$. To ensure the final dataset is not only locally diverse but also globally balanced, we compute a global diversity score $D_{\text{global}}(s)$ for each sample $s$ against the entire pool:

$$D_{\text{global}}(s) = 1 - \frac{1}{|\mathcal{S}| - 1} \sum_{s' \in \mathcal{S}, s' \neq s} \cos(\mathbf{e}_s, \mathbf{e}_{s'}). \tag{2}$$

Finally, we rank all sentences using a joint score:

$$S(s) = Q(s) + D_{\text{global}}(s), \tag{3}$$

and select the top scoring samples (i.e. 83,184) as our final synthetic dataset, CommonSyn. This global ranking process balances quality and semantic coverage across the entire dataset, encouraging the model to learn not only plausible, but also diverse sentence patterns.

We also tried to cluster on the gradients of training examples on the target model and sampled data from rarest clusters to maximize the diversity of training data influences to the model (Pan et al., 2024; Jung et al., 2025). However, empirical results (Table 7) indicated that even combining gradient with $Q(s)$ does not yield a better Q-D trade-off than the proposed embedding-based method. Furthermore, gradients is model-dependent and computationally expensive to compute (computed over all model parameters). Further implementation details are in Appendix A.

| Model | Dataset | Quality Metrics (↑) | | | Diversity Metrics (↑) | | | |
|---|---|---|---|---|---|---|---|---|
| | | Win-Tie | Cov. | **Overall** | S-BLEU3$^\dagger$ | S-BLEU4$^\dagger$ | Vendi | **S-Cos**$^\dagger$ |
| Llama-3.1-8B-Inst | Vanilla | 19.0 | 84.7 | 16.1 | 73.4 | 80.3 | 22.8 | 35.1 |
| | CommonGen | 31.7 | 85.8 | 27.2 | 76.4 | 84.7 | 21.2 | 30.7 |
| | **CommonSyn** | **47.3**$^{**}$ | **94.5** | **44.7** | 75.5 | 82.9 | 22.7 | 34.2 |
| Qwen-2.5-7B-Inst. | Vanilla | 42.9 | 92.3 | 39.6 | 39.2 | 44.1 | 15.8 | 16.3 |
| | CommonGen | 32.5 | 88.6 | 28.8 | 73.2 | 81.8 | 21.4 | 31.2 |
| | **CommonSyn** | **48.4**$^{**}$ | **93.0** | **45.0** | **73.7** | 81.0 | **22.5** | **33.7** |
| Qwen-2.5-14B-Inst. | Vanilla | 45.2 | 95.9 | 43.4 | 42.4 | 47.5 | 16.5 | 18.6 |
| | CommonGen | 38.5 | 92.9 | 35.8 | 70.8 | 79.6 | 20.9 | 29.7 |
| | **CommonSyn** | **52.0**$^{**}$ | **96.0** | **49.9** | **73.2** | **80.6** | **22.1** | **32.6** |
| Gemma-3-4B-IT | Vanilla | 28.7 | 88.4 | 25.3 | 37.7 | 42.3 | 16.1 | 17.4 |
| | CommonGen | 28.7 | 91.0 | 26.1 | 74.9 | 83.2 | 21.2 | 30.6 |
| | **CommonSyn** | **41.2**$^{**}$ | **95.5** | **39.4** | **75.4** | 82.6 | **22.4** | **33.4** |
| Gemma-3-12B-IT | Vanilla | 43.2 | 94.0 | 40.6 | 31.7 | 35.7 | 14.7 | 13.6 |
| | CommonGen | 35.9 | 94.1 | 33.8 | 72.2 | 80.4 | 20.5 | 28.7 |
| | **CommonSyn** | **51.1**$^{**}$ | **97.4** | **49.8** | **73.7** | **81.2** | **21.8** | **31.7** |
| Gemma-3-27B-IT | Vanilla | 31.9 | 86.7 | 27.7 | 27.0 | 30.2 | 15.0 | 14.7 |
| | CommonGen | 40.3 | 95.6 | 38.5 | 70.7 | 79.2 | 20.3 | 28.1 |
| | **CommonSyn** | **53.6**$^{**}$ | **97.9** | **52.5** | **71.7** | **79.2** | **21.4** | **30.7** |

Table 2: Main results for the top-6 performing models. When fine-tuned on our proposed dataset (**CommonSyn**), we observe consistent improvements achieving the best balance between quality and diversity, surpassing both the vanilla model (prior to fine-tuning) and human-annotated CommonGen data ($^{**}$: $p < 0.01$). The complete results are shown in Table 14. Diversity metrics marked with † are inverted by subtracting from 1.

| Method | Quality (↑) | | | Diversity (↑) | | |
|---|---|---|---|---|---|---|
| | Win-Tie | Cov. | **Overall** | S-B4$^\dagger$ | Vendi | **S-Cos**$^\dagger$ |
| ***Baselines*** | | | | | | |
| CommonGen | 31.7 | 85.8 | 27.2 | 84.7 | 21.2 | 30.7 |
| Prismatic (Jung et al., 2025) | 44.2 | 91.6 | 40.4 | 82.9 | 21.5 | 31.1 |
| ***Sentence Generation Methods*** | | | | | | |
| Few-shot ($D_{ms}$) | 44.6 | 90.5 | 40.4 | **85.5** | **23.4** | **36.3** |
| Dynamic ($D_{dyn}$) | **53.1** | 90.9 | **48.3** | 68.5 | 19.1 | 24.6 |
| CoT ($D_{CoT}$) | 43.4 | 95.0 | 41.2 | 80.6 | 21.9 | 32.0 |
| ***Pooled Candidates*** | | | | | | |
| 2 best ($D_{dyn} \cup D_{ms}$) | 50.0 | **95.3** | 47.7 | 75.7 | 21.0 | 30.0 |
| All ($D_{dyn} \cup D_{ms} \cup D_{CoT}$) | 50.0 | 94.4 | 47.2 | 78.8 | 21.6 | 31.4 |
| ***Selection Strategy*** | | | | | | |
| **CommonSyn** | 47.3 | 94.5 | 44.7 | 82.9 | 22.7 | 34.2 |

Table 3: Comparison of different data sources and selection strategies on Llama-3.1-8B. **CommonSyn** effectively balances quality and diversity, maintaining high Coverage. Metrics marked with † are inverted (by subtracting from 1) such that higher values indicate better performance.

# 4 Experiments

## 4.1 Evaluation Protocol

To evaluate the usefulness of CommonSyn, we fine-tune multiple LLMs on it and measure the improvements of Q-D of the generated sentences. Specifically, we conduct parameter-efficient Supervised Fine-Tuning (SFT) under unsloth (Daniel Han and team, 2023) with LoRA for all models. We require an LLM to generate each sentence one at a time in a CommonSyn training instance given the concept set for that instance and use the next token prediction as the training objective for SFT. We use the zero-shot prompt shown in Table 9 for this purpose. Multiple quality and diversity metrics are applied in our evaluations.

**Quality Metrics.** Following the standard CommonGen protocol (Lin et al., 2020), we assess quality using three metrics. (i) **Coverage:** The percentage of test examples where the generated sentences include all of the input concepts. (ii) **Win/Tie Rate:** We use GPT-4o as a judge to compare model outputs against human-written references, where the model outputs and human-written references are randomly shuffled during comparisons to guarantee fairness. The prompt (see Appendix F) instructs the LLM judge to select the sentence that is more natural and coherent and satisfies the length restriction. (iii) **Overall:** Defined as the product of Coverage and Win/Tie Rate, serving as our primary quality metric.

**Diversity Metrics.** To measure the lexical and semantic variation of the generated outputs, we use the following three metrics. (i) **Self-BLEU** (Zhu et al., 2018) measures the $n$-gram overlap between the generated sentences. We used self-BLEU-3/4 (i.e. $n = 3, 4$) in our experiments. (ii) **Self-CosSim** (Cox et al., 2021) is the average pairwise

| Task | Model | Quality (↑) | Diversity (↑) | | |
|---|---|---|---|---|---|
| | | Win-Tie | S-B4$^\dagger$ | Vendi | S-Cos$^\dagger$ |
| ComVE | Vanilla | 75.3 | 77.3 | 17.7 | 29.8 |
| | CommonGen | 41.7 | **86.3** | **21.6** | **44.9** |
| | **CommonSyn** | **78.9** | 85.9 | 18.1 | 30.0 |
| $\alpha$-NLG | Vanilla | 54.6 | 83.5 | 18.8 | 24.2 |
| | CommonGen | 25.9 | 81.3 | **22.6** | **35.3** |
| | **CommonSyn** | **61.1** | **87.9** | 19.6 | 25.7 |
| ROCStories | Vanilla | 87.2 | 79.3 | 23.9 | 38.8 |
| | CommonGen | 43.0 | **95.0** | **29.2** | **54.1** |
| | **CommonSyn** | **88.6** | 90.8 | 24.1 | 38.6 |

Table 4: Cross-task generalization on unseen generative commonsense tasks. Models fine-tuned on human data (*CommonGen*) suffer from severe **quality drop**. In contrast, model fine-tuned on **CommonSyn** generalises well, outperforming vanilla baselines while maintaining diversity. Metrics marked with † are inverted.

cosine similarity of sentence embeddings. We subtract the self-BLEU and self-CosSim scores by 1, such that higher scores indicate greater pairwise diversity. (iii) **Vendi Score (VS)-embed** (Friedman and Dieng, 2023) is the exponential of Shannon's entropy over the eigenvalues of the pairwise similarity matrix of a set of sentences computed using sentence embeddings. We use the SimCSE (Gao et al., 2021) for both Self-CosSim and VS-embed for consistency. Following (Zhang et al., 2025) we use Self-CosSim as our primary diversity metric because of its reliability.

### 4.2 Main Results

We fine-tune several LLMs on CommonSyn and compare them against vanilla (zero-shot) and CommonGen-finetuned baselines as shown in Table 2 (see Table 14 for additional results). Models trained on CommonSyn achieve significantly higher **Win-Tie** rates and **Coverage** than the vanilla and CommonGen baselines (significant at $p < 0.01$ according to the bootstrap sampling test (Hesterberg, 2011)). For example, although Qwen-2.5-7B/14B already has strong performance on quality metrics, CommonSyn boosts their **Overall** score (+5.4) and doubling the diversity scores (S-Cos 16.3 to 33.7), showing that CommonSyn successfully injects semantic diversity without degrading commonsense quality.

A notable exception in diversity is observed with Llama-3.1-8B-Inst, where the vanilla model exhibits high diversity scores (S-Cos 35.1) but extremely low quality on both **Coverage** and **Win-Tie**. As further analysed later in § 4.6, this stems from verbose generations that fail to reliably cover all of the given input concepts. In contrast, CommonSyn improves the **Overall** score (from 16.1 to 44.7), while not harming the diversity too much. This confirms that CommonSyn jointly optimises the Q-D objective. Moreover, CommonSyn shows consistent gains across model sizes ranging from 4B to 27B parameters. The 27B model fine-tuned on CommonSyn achieves the highest reported **Win-Tie** rate, demonstrating that even a capable large model can be further improved for quality using diversity-aware synthetic fine-tuning.

### 4.3 Sentence Generation Strategies

In Table 3, we compare the different sentence generation methods, pooled candidates and the selection strategy described in § 3.3 by fine-tuning Llama-3.1-8B-Instruct. We observe that each individual generation strategy has different Q-D performance. Although **Dynamic Few-Shot Generation** ($D_{dyn}$) achieves the highest **Win-Tie** rate, it shows severe mode collapse, resulting in the lowest diversity. On the other hand, **Multi-sentence Few-shot Generation** ($D_{ms}$) produces the most diverse outputs but suffers from both lower quality and concept set Coverage. Although **CoT-guided Generation** ($D_{CoT}$) has a moderate Overall Score, it reports the highest **Coverage** (95.0%) among all sentence generation methods due to explicit reasoning.

Given the complementary strengths of the three sentence generation strategies, we found that simply taking the union of their generated sentence sets to be suboptimal, as shown in Table 3 under **Pooled Candidates**. Although pooling improves **Coverage**, pooled data underperforms $D_{ms}$ in diversity and $D_{dyn}$ in quality. This confirms that naive merging introduces noise that reduces the benefits of high-potential candidates, emphasising the importance of sentence selection.

CommonSyn outperforms Prismatic Synthesis (same number of samples) (Jung et al., 2025), a gradient-based synthetic data generation method which generates and selects candidates by clustering samples in the gradient space of the LLM and up-samples according to a sparsity metric defined over the sentence clusters. Furthermore, Prismatic Synthesis needs proxy models and computing the gradients over all the samples iteratively, which is computationally expensive. In contrast, our proposal remains more computationally attractive.

### 4.4 Downstream Evaluation

Recall that CommonSyn was created using seed concepts from CommonGen. Therefore, it remains

| **Dataset** | **CSQA** | **CSQA2** | **PIQA** | **Avg.** |
|---|---|---|---|---|
| Vanilla | 70.3 | 51.8 | 69.2 | 63.8 |
| CommonGen | 53.6 | **60.4** | 67.0 | 60.1 |
| **CommonSyn** | **70.8** | 49.0 | **71.3** | 63.7 |

Table 5: Zero-shot accuracy (%) on discriminative commonsense QA tasks. Training on human-annotated data causes **catastrophic forgetting** (avg. $-3.7$ pp), whereas **CommonSyn** preserves general reasoning capabilities (avg. $-0.1$ pp).

| Method | Unique Concepts | % Unseen Concepts | % Unseen Triples |
|---|---|---|---|
| Human (CommonGen) | 3,677 | 75.36 | 100 |
| Direct Generation | 805 | 52.55 | 96.65 |
| Augment (Add-$C$) | 3,791 | 72.80 | 99.65 |
| **2-Seed Expansion** | **4,260** | **75.28** | **99.62** |

Table 6: Comparison of concept set construction strategies ($N = 20k$). **2-Seed Expansion** gets the highest number of unique concepts and maintains high novelty relative to the test set, indicating no data leakage.

a question whether CommonSyn could improve performance in other GCR tasks. To evaluate the zero-shot generalisability of CommonSyn, we fine-tuned Llama-3.1-8B-Instruct on CommonSyn and evaluated it on multiple generative and question answering tasks that require commonsense reasoning.

**Generative Tasks:** Three GCR benchmarks are selected for this evaluation: (a) **ComVE** (Wang et al., 2020) (generate explanations for counterfactual statements), (b) $\alpha$-**NLG** (Bhagavatula et al., 2020) (perform abductive reasoning to infer plausible hypotheses), and (c) **ROCStories** (Mostafazadeh et al., 2016) (complete a story by generating a coherent ending). The zero-shot inference prompts are shown in § D.2.

Following the CommonGen evaluation protocol, we limit the output length and use GPT-4o to judge the pairwise quality between model generations and human references by the **Win-Tie** rate. As shown in Table 4 models fine-tuned on **CommonGen** generalise poorly to other tasks. Moreover, the quality of the generations drop significantly compared to the vanilla baseline (e.g., the **Win-Tie** rate nearly halves on ComVE: $75.3 \rightarrow 41.7$). Although diversity metrics appear to be high, we find many hallucinations and commonsense violations as further discussed in § D.1.

In contrast, models fine-tuned on **CommonSyn** generalises well to other tasks by consistently outperforming both the vanilla baseline and the CommonGen-finetuned models in terms of quality. Furthermore, simultaneous improvements for both quality and diversity can be observed over the vanilla model. These results suggest that CommonSyn improves the GCR capabilities not only for CommonGen, but also for other tasks.

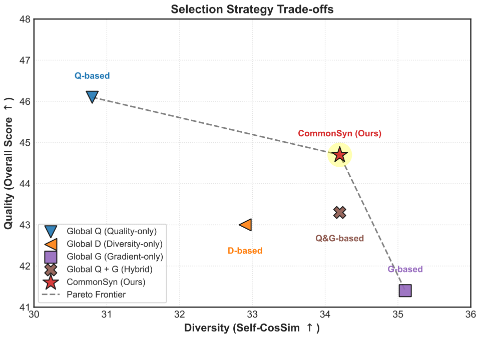

Figure 2: Trade-off between Quality and Diversity. The dashed line indicates the Pareto frontier. **CommonSyn** (Red Star) strictly outperforms the gradient-hybrid method *Q&G-based* (Brown Cross) by achieving higher quality at the same diversity level.

| **Strategy** | **Quality (↑)** | | | **Diversity (↑)** | | |
|---|---|---|---|---|---|---|
| | Win-Tie | Cov. | **Overall** | S-B4$^{\dagger}$ | Vendi | S-Cos$^{\dagger}$ |
| Q-based | **49.0** | 94.1 | **46.1** | 78.5 | 21.4 | 30.8 |
| D-based | 45.6 | 94.3 | 43.0 | 80.6 | 22.2 | 32.9 |
| G-based | 43.9 | 94.4 | 41.4 | **83.8** | **23.0** | **35.1** |
| Q&G-based | 45.7 | **94.7** | 43.3 | 81.8 | 22.7 | 34.2 |
| **CommonSyn** | 47.3 | 94.5 | 44.7 | 82.9 | 22.7 | 34.2 |

Table 7: Comparison of selection strategies. **CommonSyn** outperforms the gradient-based hybrid (*Q&G-based*) in quality while matching its diversity, validating the effectiveness of our 2-stage selection.

**Commonsense QA Tasks:** We further evaluate the generalisability of CommonSyn finetuned models on three commonsense QA tasks: CSQA (Talmor et al., 2019), CSQA2 (Talmor et al., 2021) and PIQA (Bisk et al., 2020). As shown in Table 5, the CommonGen-finetuned model exhibits average accuracy dropping by 3.7 percentage points (from 63.8 to 60.1).

In contrast, the CommonSyn-finetuned model not only avoids this degradation—it even shows slight improvements on CSQA (+0.5 pp) and PIQA (+2.1 pp), resulting in near-identical average performance to the vanilla baseline (63.7 vs. 63.8).

## 4.5 Ablation Study

**Concept Set Construction Strategy** We compare our **2-Seed Expansion** strategy against CommonGen and two baselines: **Direct Generation**

| Input Concepts | Human Reference | Vanilla | CommonGen | CommonSyn (Ours) |
|---|---|---|---|---|
| *dog, throw, frisbee, catch* | My dog likes to catch frisbees after I throw them. | As the sun was setting... a energetic dog awaited its owner's signal to throw the frisbee... | A golden colored dog **throws a frisbee** and catches it. ✗(Logic) | I **throw** the frisbee, and my dog leaps to **catch** it mid-air. |
| *ride, dog, run, shoot, horse* | A dog runs by while the man rides a horse and shoots. | As the rodeo rider rode his horse at full speed, he expertly steered it... (Misses: dog, shoot) | ...shoots a watermelon with a water pistol **on a dog**. ✗(Nonsense) | She **rode** the **horse** past a field where her **dog ran** and someone **shot** at a deer. |
| *instruction, tie, neck, give* | I had to give instructions to tie the tie around her son's neck. | To properly tie a neck tie, follow the instructions carefully... | **tied up in knots how to** give instructions for tying a tie. ✗(Grammar) | In class, the teacher **gives** detailed **instruction** on how to properly **tie** a neck **tie**. |

Table 8: A comparison of sentences generated by different methods. Although **CommonSyn** generates concise, logically sound, and grammatically correct sentences that fully cover the input concepts, other approaches often suffer from verbosity (Vanilla), commonsense violations (CommonGen), or incorrect grammar.

(Directly require LLM to generate the concept set) and **Augment** (Add-$C$), which augments an existing concept set first by adding $C$ related concepts and then randomly removing $C$ concepts from it to form a new set. We sample 20k concept sets for each method and report Unique Concepts and Unseen rates relative to the test set. As shown in Table 6, **Direct Generation** suffers from **low diversity** in the generated concept sets, covering only 805 unique concepts as the model regresses to high-frequency generic terms. Although **Augment** improves coverage, it remains bounded by the original distribution. In contrast, our **2-Seed** generates more unique concepts than the human-crafted CommonGen dataset and maintains a high rate of unseen concept composition (unique concept, concept-triples). This confirms that two seeds strategy effectively overcomes the coverage of human-written sentences while preventing data leakages.

Furthermore, CommonSyn shows stable performance across different concept set sizes, suggesting high data efficiency. In the Table 10, around 20k concept sets are sufficient to saturate performance while remaining cost-effective. A detailed analysis of dataset scaling is provided in Appendix B.

**Data Selection Strategies** We compare different data selection methods considering quality, diversity, and gradient and its hybrid with quality approach. The number of samples in the comparison is unique. As shown in Figure 2, these strategies occupy distinct positions in the performance landscape. The Quality-based (Q-based) selection strategy, which selects the coreset based on the score given by GPT-4o scorer as mentioned in § 3.4, maximises quality but suffers from low diversity. The Gradient-based (G-based) strategy conducts $k$-means clustering using the loss gradient obtained from a proxy model for each sentence and then samples according to cluster sparsity (Jung et al., 2025). G-based prioritises diversity over quality. *Q&G-based* improves quality but still underperforms CommonSyn. Furthermore, gradient-based methods require expensive gradient embeddings (e.g., for Llama-3.1-8B), while CommonSyn achieves a better trade-off at a lower computational cost. Among all the data selection strategies, CommonSyn shows the optimal balance on the Pareto frontier.

### 4.6 Case Study

To show the qualitative differences between models, we compare generations from the vanilla model, CommonGen-finetuned, and our **CommonSyn** in Table 8. We observe that the vanilla model omits input concepts (e.g., missing "dog" and "shoot" in the second example) and tends to produce verbose text that deviates from the concise style expected in CommonGen. The CommonGen-finetuned model arranges the concepts into implausible scenarios (e.g., "shooting a watermelon with a water pistol on a dog"), violating commonsense constraints. In contrast, **CommonSyn** consistently integrates all input concepts into coherent, natural, and logically plausible sentences. For example, in the first example, it correctly assigns agency: the human *throws* the frisbee, and the dog *catches* it—aligning both with physical plausibility and task expectations. Additional examples are provided in Table 11.

## 5 Conclusion

We created CommonSyn, a synthetic dataset to jointly improve both quality and diversity in GCR. Models fine-tuned on CommonSyn demonstrate Patero Q-D improvements and generalise to other GCR tasks beyond CommonGen.

# 6 Limitations

Despite the effectiveness of CommonSyn, we acknowledge the following limitations. First, our dataset is currently restricted to English, a morphologically limited language. Although extending CommonSyn to multilingual settings is a desirable direction, we are constrained by the scarcity of high-quality human reference benchmarks in other languages. In particular, CommonGen (Lin et al., 2020), ComVE (Wang et al., 2020), and $\alpha$-NLG (Bhagavatula et al., 2020) datasets are specifically designed to evaluate diversified commonsense reasoning only in English.

Furthermore, the quality of synthetic data depends on the quality of the generator. Our method relies on Qwen2.5-72B-Instruct for data synthesis and a separate model (Gemini-2.5-Flash) for quality scoring. Therefore, the final dataset is inherently bounded by the capabilities and biases of these teacher models and a better generator might lead to higher quality synthetic data, such as by using GPT-4o to generate sentences. However, we used GPT-4o as an evaluator to compare the model outputs and human references. Therefore, we cannot directly use it as a generator or quality scorer when creating synthetic data.

We also acknowledge potential intrinsic biases in using GPT-4o as an automated evaluator. Although our evaluation framework is adaptable to other strong models, we selected GPT-4o for its superior performance in NLG evaluations. Given that valid GCR outputs can be open-ended, we rely on Win-Tie rates against human references to ensure robustness. Although we believe this offers a scalable alternative to human annotation, verifying our findings with other comparable or superior LLMs remains an important direction for future work.

Our global selection stage requires computing pairwise cosine similarities for the entire candidate pool to determine global diversity scores. Although we argue that this is more efficient than gradient-based methods, the computational complexity of pairwise comparisons can still become a bottleneck when scaling to significantly larger candidate pools (e.g., millions of samples).

Our primary evaluation focuses on CommonGen and related GCR tasks. Although we demonstrate generalisation ability to tasks like ROCStories, our **2-seed concept expansion** strategy is explicitly designed for *concept-to-text* generation tasks, where the input is a discrete set of keywords. This module is not directly applicable to standard sequence-to-sequence tasks (e.g., summarisation, translation, or open-ended story generation) where the input typically consists of complete sentences. Applying our method to such tasks would require redesigning the question generation strategies.

# 7 Ethical Consideration

All experiments conducted in this study use publicly available datasets such as CommonGen, ComVE, and $\alpha$-NLG. To the best of our knowledge, no personally identifiable information has been included in those datasets and no ethical issues have been reported.

Our study introduces *CommonSyn*, a synthetic dataset for commonsense reasoning generated using LLMs and it is noteworthy that LLMs have been reported to encode social biases such as gender or racial biases (Kaneko and Bollegala, 2021; Nangia et al., 2020; Kaneko et al., 2022). Although we evaluated quality and diversity of the generations made by LLMs and automatic metrics in this work, we defer a systematic evaluation of social or representational biases in the generated content to future work. We emphasise the importance of evaluating social biases in our synthetic dataset before it is deployed in NLG applications used by human users. We release our dataset with this disclaimer and encourage its users to apply appropriate filtering, debiasing, or mitigation strategies as needed.

# References

Zachary Ankner, Cody Blakeney, Kartik Sreenivasan, Max Marion, Matthew L Leavitt, and Mansheej Paul. 2024. Perplexed by perplexity: Perplexity-based data pruning with small reference models. *arXiv preprint arXiv:2405.20541*.

André Bauer, Simon Trapp, Michael Stenger, Robert Leppich, Samuel Kounev, Mark Leznik, Kyle Chard, and Ian Foster. 2024. Comprehensive exploration of synthetic data generation: A survey. *arXiv preprint arXiv:2401.02524*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. *Preprint*, arXiv:1908.05739.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I Abdin. 2024a. On the diversity of synthetic data and its impact on training large language models. *arXiv preprint arXiv:2410.15226*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and 1 others. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024b. Self-play fine-tuning convertsweak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 6621–6642.

Samuel Rhys Cox, Yunlong Wang, Ashraf Abdul, Christian Von Der Weth, and Brian Y. Lim. 2021. Directed diversity: Leveraging language embedding distances for collective creativity in crowd ideation. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–35.

Wanqing Cui, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. More: Multi-modal retrieval augmented generative commonsense reasoning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1178–1192.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Zhiwei Deng, Tao Li, and Yang Li. 2024. Influential language data selection via gradient trajectory pursuit. *arXiv preprint arXiv:2410.16710*.

Yunzhen Feng, Elvis Dohmatob, Pu Yang, Francois Charton, and Julia Kempe. 2025. Beyond model collapse: Scaling up with synthesized data requires verification. In *The Thirteenth International Conference on Learning Representations*.

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

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, YK Li, and 1 others. 2024. Deepseek-coder: When the large language model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*.

Tim Hesterberg. 2011. Bootstrap. *Wiley Interdisciplinary Reviews: Computational Statistics*, 3(6):497–526.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Yiming Huang, Xiao Liu, Yeyun Gong, Zhibin Gou, Yelong Shen, Nan Duan, and Weizhu Chen. 2025. Key-point-driven data synthesis with its enhancement on mathematical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24176–24184.

EunJeong Hwang, Veronika Thost, Vered Shwartz, and Tengfei Ma. 2023. Knowledge graph compression enhances diverse commonsense generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 558–572, Singapore. Association for Computational Linguistics.

Jaehun Jung, Seungju Han, Ximing Lu, Skyler Hallinan, David Acuna, Shrimai Prabhumoye, Mostafa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Yejin Choi. 2025. Prismatic synthesis: Gradient-based data diversification boosts generalization in llm reasoning. *arXiv preprint arXiv:2505.20161*.

Masahiro Kaneko and D Bollegala. 2021. Unmasking the mask - evaluating social biases in masked language models. *National Conference on Artificial Intelligence*, pages 11954–11962.

Masahiro Kaneko, Aizhan Imankulova, Danushka Bollegala, and Naoaki Okazaki. 2022. Gender bias in masked language models for multiple languages. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2740–2750, Stroudsburg, PA, USA. Association for Computational Linguistics.

Vishal Kaushal, Rishabh Iyer, Suraj Kothawade, Rohan Mahadev, Khoshrav Doctor, and Ganesh Ramakrishnan. 2019. Learning from less data: A unified data subset selection and active learning framework for computer vision. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1289–1299. IEEE.

Cheng Li, Mengzhuo Chen, Jindong Wang, Sunayana Sitaram, and Xing Xie. 2024. Culturellm: Incorporating cultural differences into large language models. *Advances in Neural Information Processing Systems*, 37:84799–84838.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Chenzhengyi Liu, Jie Huang, Kerui Zhu, and Kevin Chen-Chuan Chang. 2023. DimonGen: Diversified generative commonsense reasoning for explaining concept relationships. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 4719–4731, Toronto, Canada. Association for Computational Linguistics.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and 1 others. 2024. Best practices and lessons learned on synthetic data. *arXiv preprint arXiv:2404.07503*.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477.

Brando Miranda, Alycia Lee, Sudharsan Sundar, Allison Casasola, and Sanmi Koyejo. 2023. Beyond scale: The diversity coefficient as a data quality metric for variability in natural language data. *arXiv preprint arXiv:2306.13840*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Ranjita Naik, Varun Chandrasekaran, Mert Yuksekgonul, Hamid Palangi, and Besmira Nushi. 2023. Diversity of thought improves reasoning abilities of llms. *arXiv preprint arXiv:2310.07088*.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xingyuan Pan, Luyang Huang, Liyan Kang, Zhicheng Liu, Yu Lu, and Shanbo Cheng. 2024. G-dig: Towards gradient-based diverse and high-quality instruction data selection for machine translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15395–15406.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. 2023. Trak: Attributing model behavior at scale. In *International Conference on Machine Learning (ICML)*.

Yulei Qin, Yuncheng Yang, Pengcheng Guo, Gang Li, Hang Shao, Yuchen Shi, Zihan Xu, Yun Gu, Ke Li, and Xing Sun. 2024. Unleashing the power of data tsunami: A comprehensive survey on data assessment and selection for instruction tuning of language models. *arXiv preprint arXiv:2408.02085*.

Ozan Sener and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2021. CommonsenseQA 2.0: Exposing the limits of AI through gamification. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. 2023. D4: Improving llm pretraining via document de-duplication and diversification. *Advances in Neural Information Processing Systems*, 36:53983–53995.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. Semeval-2020 task 4: Commonsense validation and explanation. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 307–321.

Zaitian Wang, Jinghan Zhang, Xinhao Zhang, Kunpeng Liu, Pengfei Wang, and Yuanchun Zhou. 2025. Diversity-oriented data augmentation with large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22265–22283, Vienna, Austria. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2024. Lamini-lm: A diverse herd of distilled models from large-scale instructions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 944–964.

Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. 2022. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *The Eleventh International Conference on Learning Representations*.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. 2023. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *ICLR*.

Wenhao Yu, Chenguang Zhu, Lianhui Qin, Zhihan Zhang, Tong Zhao, and Meng Jiang. 2022. Diversifying content generation for commonsense reasoning with mixture of knowledge graph experts. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*, pages 1–11.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in neural information processing systems*, 36:55734–55784.

Tianhui Zhang, Bei Peng, and Danushka Bollegala. 2024. Improving diversity of commonsense generation by large language models via in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9226–9242, Miami, Florida, USA. Association for Computational Linguistics.

Tianhui Zhang, Bei Peng, and Danushka Bollegala. 2025. Evaluating the evaluation of diversity in commonsense generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24258–24275, Vienna, Austria. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

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 Implementation Details

For our synthetic data, we use Qwen2.5-72B-Insturct with $T = 1.0$ to generate both concept sets and the corresponding sentence sets with the three sentence generation strategies for reproducibility. The 2-Seed Expansion strategy uses two randomly selected concepts per seed to generate novel combinations. To prevent potential data leakage, we ensure that concepts from CommonGen test sets are excluded from the training set, ensuring a 99.6% unseen concept-triple rate (see Table 6). For Dynamic Few-shot, we use 5 exemplars with temperature set to $T = 1.0$.

We selected a total of 11 widely used instruct-tuned models for fine-tuning and evaluation: Llama-3.1-8B, 3.2-1B, 3B; Qwen-2.5-1.5B, 3B, 7B, 14B; Gemma3-1B, 4B, 12B, 27B. We fine-tuned these models under SFT. All models are fine-tuned with Unsloth (Daniel Han and team, 2023) using LoRA with rank $r = 96$ and $\alpha = 192$. The prompt for training samples is shown in Table 9. For inference, we use a temperature of $T = 1.0$ to balance diversity and coherence (higher $T$ encourages more diverse outputs). For data selection, we use **Gemini-2.5-Flash** as the quality scorer because of its cost-efficiency.

To compare our method with gradient-based data selection on the Llama-3.1-8B model, we follow the procedure of Jung et al. (2025). Specifically, we compute gradients of candidate data samples on the model with respect to all trainable parameters, flatten and concatenate them into a single vector. Then we randomly project the resulting high-dimensional gradients into a 1024-dimensional space using TRAK (Park et al., 2023). We use K-means clustering algorithm and sample the samples from the top-50% rarest clusters to ensure the diversity of training data influences. In practice, we cluster the training data into 831 clusters.

For the final evaluation, following the evaluation setting of (Lin et al., 2020), we use GPT-4o to evaluate the preference between generated sentences and human-written references. In both cases, the temperature is set to $T = 0$ to obtain reproducible generations. We used A100 GPUs for all experiments.

| Role | Prompt |
|---|---|
| System | Given several keywords, generate one coherent sentence that contains all the required keywords using background commonsense knowledge: |
| User | {Concept Set} |
| Assistant | {Sentence} |

Table 9: Prompts used for training samples for CommonSyn.

# B Number of Concept Sets

| Num of Concept Sets | Quality (↑) | | | Diversity (↑) | | |
|---|---|---|---|---|---|---|
| | Win-Tie | Cov. | **Overall** | S-B4$^\dagger$ | Vendi | S-Cos$^\dagger$ |
| 5k | 46.0 | **94.4** | 43.4 | 82.2 | 22.4 | 33.5 |
| 10k | 46.2 | 94.3 | 43.6 | 82.7 | 17.3 | 34.3 |
| 15k | 45.0 | **94.4** | 42.5 | 82.6 | 22.5 | 33.9 |
| **20k** | **46.6** | 94.0 | **43.8** | **82.8** | **22.8** | **34.6** |

Table 10: Performance obtained with numbers of unique concept sets (fixed total sentence sample size = 40k). Consistently high performance is seen across concept set sizes, indicating high data efficiency of our proposed method.

We compare how the diversity of input scenarios (number of unique concept sets) affects performance. We fix the total dataset size at 40k sentence samples and vary the number of unique concept sets from 5k to 20k (implying that fewer sets have more sentences per set, and vice versa).

Table 10 shows that the performance is stable across different scales. Increasing the unique concept sets from 5k to 20k yields only slight improvement in **Overall** (quality) and **self-CosSim** (diversity). This suggests that **CommonSyn** captures the core underlying logic of commonsense generation efficiently, without requiring a large possible concept combination. In addition, ∼20k concept sets are sufficient to saturate the model's capability for this task, showing the concept set size in our final dataset as a cost-effective spot.

# C Case Study for CommonGen Task

In the Table 11, we compare the multiple generations from vanilla CommonGen and CommonSyn-finetuned model using Qwen2.5-7B-Instruct model. For each concept set, we sample four independent generations from each model to illustrate output diversity, concept coverage, and commonsense plausibility. In the result, we can find although vanilla model can generate plausible sentences, the sentences it generates are highly redundant. The sentences that are generated by CommonGen-finetuned model sometimes will are commonsense-conflicting ("Dog throws the Frisbee for another dog to catch."). However, the model fine-tuned on our CommonSyn generates both high quality and diversity sentences.

# D Downstream Tasks Evaluation

In this section, we provide details of our **zero-shot** experiments on unseen generative tasks, including qualitative case studies and the inference prompts used in evaluation (see § D.2 for prompt templates). There are three commonsense reasoning tasks we choose for generative tasks:

- ComVE (Wang et al., 2020) aims to generate an short explanation given a counterfactual statement to explain why the thing described by statement can not happen in real world.
- $\alpha$-NLG (Bhagavatula et al., 2020) is the task of generating a valid hypothesis about the likely explanations to partially observe pas and future. The input is two observations with ordering and the hypothesis generated by model should design a scenario to connect the observations.
- ROCStories (Mostafazadeh et al., 2016) requires a model to write the plausible ending to a four-sentence daily story.

## D.1 Case Study on More Tasks

To better understand the nature of the *quality drop* observed in models fine-tuned on human-annotated CommonGen data, we conduct a qualitative analysis of model outputs across three diverse generative commonsense tasks: **ComVE**, **$\alpha$-NLG**, and **ROCStories**. Table 12 presents a comparison across three tasks: ComVE, $\alpha$-NLG, and ROCStories.

As shown in Table 12, although the CommonGen-finetuned model sometimes achieves high lexical diversity (Table 4), its generations often suffer from hallucination, logical inconsistency—issues that degrade output quality despite surface-level fluency.

For example, in the **ComVE** task—which requires explaining why a given statement is counterfactual—the CommonGen model generates: "The image shows sugar is added to make coffee sour."

| **Input Concepts** | **Vanilla** | **CommonGen** | **CommonSyn (Ours)** |
|---|---|---|---|
| *take, shot, goal, player* | 1. The player took a shot with the intention of scoring the goal in the upcoming match.<br>2. The player takes a shot with the intention of scoring the goal in the ongoing soccer match.<br>3. The goal keeper took a daring shot at goal, hoping to catch the player off guard.<br>4. The player takes a shot with the intention of scoring the goal in the ongoing soccer match. | 1. soccer player takes a shot on goal during their match .<br>2. football player takes a shot on goal during a friendly match .<br>3. soccer player taking a shot for the goal during the match .<br>4. football player takes a shot on goal during a training session . | 1. The player takes a shot to score the final goal in the championship.<br>2. During the intense match, the player took a bold shot toward the goal.<br>3. The soccer player took a powerful shot to score the goal.<br>4. The soccer player takes a fierce shot towards the goal, hoping to score. |
| *pan, stove, cook, food* | 1. She can pan fry the chicken on the stove to cook the delicious food.<br>2. In the morning, I like to cook delicious food on a stove using a pan.<br>3. She uses a stove to cook food in a pan.<br>4. She uses a pan on the stove to cook food for dinner. | 1. A young boy cooking food with a pan on the stove.<br>2. A man is cooking food in a pan on a stove.<br>3. A man is cooking food in a pan on a stove.<br>4. a man cooks with a frying pan on the stove. | 1. In the morning, he stands at the stove to cook breakfast in a pan.<br>2. I use a pan on my stove to cook food every day.<br>3. Every Sunday, I cook a hearty meal on my pan over the stove.<br>4. He carefully cooked the pasta on the stovetop pan for his family. |
| *dog, throw, frisbee, catch* | 1. The dog happily catches the frisbee that was thrown by its owner.<br>2. The dog threw the frisbee and then eagerly waited to catch it.<br>3. The dog threw the frisbee and then eagerly waited to catch it.<br>4. The dog threw the frisbee and then patiently waited to catch it again. | 1. A young boy throwing a frisbee to his dog as he catches it.<br>2. A woman is throwing a frisbee for her dog to catch.<br>3. Two dogs play tug with a frisbee.<br>4. Dog throws the Frisbee for another dog to catch. | 1. My dog stands excitedly as I throw the frisbee; it always loves a good catch.<br>2. She throws a frisbee for her dog to catch in the park.<br>3. My dog always catches the frisbee I throw during our evening walks.<br>4. The owner throws the frisbee for his dog to catch, enjoying the sunny day. |
| *coffee, drink, newspaper, read* | 1. He likes to drink coffee while reading the newspaper in the morning.<br>2. He likes to read the newspaper while drinking coffee in the morning.<br>3. He likes to read the newspaper while drinking coffee in the morning.<br>4. He likes to drink coffee while reading the newspaper in the morning. | 1. young woman reading a newspaper while drinking coffee and smoking.<br>2. young women drinking a coffee and reading the newspaper at home.<br>3. A woman is reading a newspaper while drinking coffee.<br>4. The man is drinking coffee while reading a newspaper. | 1. She enjoys a quiet morning with a cup of coffee and the local newspaper, sipping it as she reads.<br>2. He likes to drink his coffee while reading the morning newspaper.<br>3. Every morning, he drinks a cup of coffee and reads the newspaper.<br>4. After work, I sit down to read the newspaper and drink coffee. |
| *ball, throw, pitcher, batter* | 1. The pitcher threw the ball to the batter during the baseball game.<br>2. The pitcher threw the ball to the batter during the baseball game.<br>3. The pitcher threw the ball to the batter during the baseball game.<br>4. The pitcher throws the ball to the batter during the baseball game. | 1. The pitcher has just thrown a pitch to home plate and the batter is swinging.<br>2. A batter is waiting for the ball to be thrown by the pitcher.<br>3. a ball thrown by a pitcher and the batter.<br>4. pitcher throws a ball at the batter. | 1. The pitcher throws the ball, but the batter swings and misses, watching it sail past.<br>2. The batter waited for the ball to be thrown by the pitcher.<br>3. During practice, a young batter focuses on how to react to a thrown ball by the pitcher.<br>4. The pitcher throws a curve ball past the batter. |

Table 11: Extended case study showing four sampled generations per model for each concept set. CommonGen often ignores or distorts input concepts, while Vanilla exhibits high redundancy. CommonSyn produces diverse, fluent, and concept-consistent outputs that respect real-world roles and relationships.

This response hallucinates a non-existent visual context, violating the purely textual nature of the task. In contrast, our **CommonSyn**-finetuned model correctly reasons that "sugar is typically added to make coffee sweeter, not sourer," demonstrating grounded commonsense understanding.

Similarly, in $\alpha$**-NLG**, where the goal is to abductively link two observations, the CommonGen model produces a hypothesis that directly contradicts the second observation (e.g., claiming Timmy *passed* when O2 states he *failed*). The Vanilla model, while coherent, often lacks causal depth. Only **CommonSyn** consistently generates plausible, fact-consistent explanations that respect both input observations.

These cases show that fine-tuning on human-annotated CommonGen data can induce format overfitting, leading models to prioritise stylistic similar (e.g., descriptive elaboration) over semantic fidelity. In contrast, training on **CommonSyn** shows generalizable reasoning skills and performs robustly on unseen tasks.

### D.2 Downstream Tasks Prompting

In this section, we show the generation prompts and evaluation prompts in these downstream tasks. To limit the surface diversity of sentence groups due to their length, we set the generation length to 22 for each generative task (Table 13).

In Figure 3, we present the evaluation prompts from each downstream generative task. Apart from the plausibility, the generation also needs to meet the requirements for different tasks. When evaluate the generation for each task, we also randomly shuffle the model generated sentences and the human-written references so that the ordering will not affect the LLM evaluation result.

## E Generation Prompting

In this section, we detail the prompting strategies used to construct our synthetic dataset CommonSyn. First, we expand each 2-seed concept pair into a coherent event-centric concept set using the instruction in Figure 4. Then, for each expanded concept set, we generate candidate sentences via three distinct prompting paradigms—Dynamic Few-Shot, Multi-sentence Few-Shot, and Chain-of-Thought as specified in Figure 5. All generated sentences are constrained to $\leq$ 22 words in the prompts.

## F Evaluation Prompting

In this section, we present the evaluation prompts used to assess the quality of sentences. The prompt used to evaluate sentence plausibility per concept set and return score for sentences is shown in Figure 6.

In Figure 7, the prompt is used to assess the quality of model-generated outputs against human-written references from the CommonGen test set (Lin et al., 2020). Following the official evaluation protocol from the CommonGen-Eval repository, we employ pairwise comparison prompts that instruct GPT-4o to select the better sentence based on naturalness, concept coverage, and plausibility of everyday scenarios.

## G Full Result Table

Table 14 shows the full result table for all the models we evaluate our CommonSyn synthetic data. We do experiments on 11 models and the model fine-tuned on CommonSyn consistently outperforms the vanilla models and CommonGen-finetuned models.

| **Task Input** | **Vanilla (Zero-shot)** | **CommonGen (Human)** | **CommonSyn (Ours)** |
|---|---|---|---|
| ***Task 1: ComVE*** *(Generate explanation for why the statement is false)* | | | |
| **Input:** sugar is used to make coffee sour | That's a counterfactual statement.<br>*(Refusal/Vague)* | "The image shows sugar is added to make coffee sour."<br>✗*(Hallucination: no image exists)* | Sugar is typically added to make coffee sweeter, not sourer.<br>✔*(Correct commonsense reasoning)* |
| **Input:** The inverter was able to power the continent. | It defies the fundamental laws of physics and engineering. | The inverter in the car wasn't able to power the entire continent of Europe. | The inverter, no matter how powerful, cannot power an entire continent due to its size and energy requirements. |
| ***Task 2: $\alpha$-NLG*** *(Generate a hypothesis connecting Observation 1 and 2)* | | | |
| **O1:** Timmy was running fast to class.<br>**O2:** Timmy later found out that he failed. | Timmy likely ran to class because he forgot that the exam was scheduled for a specific time and he didn't want to be late. | Because Timmy was running fast to class, he made it to his exam on time and therefore he passed.<br>✗*(Contradicts O2: he failed)* | Timmy's need to be on time for the exam, combined with his earlier desire to be punctual, suggests he likely rushed to class, possibly missing key information.<br>✔*(Plausible inference consistent with failure)* |
| **O1:** Music was Kat's life.<br>**O2:** She completed her studies with honors and is now a renowned pianist. | Kat enjoyed it so much that she played all of the time. | Music was Kat's life and she is now a renowned pianist.<br>✗*(Mere repetition, no causal link)* | Kat's dedication to music fueled her academic success, leading to her distinguished career as a pianist. |
| ***Task 3: ROCStories*** *(Generate a coherent story ending)* | | | |
| **Context:** Harriet was reading a great mystery book. She read it all evening and quietly closed the book. | As she set the book on the nightstand, her phone buzzed with an incoming text from her best friend, Emily.<br>*(Introduces irrelevant new element)* | The cover featured a silhouette of a lone person in a dense forest with a cabin to the right.<br>✗*(Describes cover, not ending!)* | Feeling the tension dissipate, Harriet leaned back into the couch, her eyes already drifting shut.<br>✔*(Natural, thematically consistent closure)* |

Table 12: Qualitative comparison on downstream generative tasks. Models fine-tuned on human-authored CommonGen data generalise poorly: they hallucinate non-existent images (ComVE), contradict input facts ($\alpha$-NLG), or generate off-topic descriptions (ROCStories). In contrast, **CommonSyn** shows robust commonsense reasoning and task adherence across unseen domains.

| **Task** | **Inference Prompt** |
|---|---|
| ComVE | Given an implausible or counterfactual statement, generate one short explanation ($\leq$ 22 words) that explains why it is implausible or counterfactual using background commonsense knowledge. |
| $\alpha$-NLG | Given an initial observation and a later observation, generate a short hypothesis ($\leq$ 22 words) that bridges Observation 1 and Observation 2 using background commonsense knowledge. |
| ROCStories | Read the following 4-sentence story context and write a short and plausible ending ($\leq$ 22 words) to this story using background commonsense knowledge. |

Table 13: Inference prompts used for zero-shot evaluation on downstream generative tasks. All outputs are constrained to $\leq$ 22 words.

**ComVE Evaluation Prompt:**

Given an implausible or counterfactual statement, we ask models to generate a short explanation of why the statement is implausible or counterfactual.
Statement: {$input}
Model A: {$candidate_A}
Model B: {$candidate_B}

Your Task: Choose the better explanation. Decide which model's output better explains why the statement is implausible.
Rules:
– A good explanation should point out the everyday commonsense reason why the statement is unrealistic.
– Prefer simple, intuitive explanations that a person would naturally give.
– Prefer explanations that highlight the obvious mismatch with real-world scenes.
– If both explanations are equally good or equally flawed, choose "tie".
Now, please output your choice ("A", "B", or "tie").

**$\alpha$-NLG Evaluation Prompt:**

Given two observations (O1 and O2) that describe the beginning and end of a short scenario, we ask models to generate a "Hypothesis" (a middle sentence) that explains what happened in between to cause the transition from O1 to O2.
Input Data: "{$input}"
Model A: "{$candidate_A}"
Model B: "{$candidate_B}"

Your Task: Choose the better hypothesis. Decide which model's output creates a more plausible and coherent story bridge between the two observations.
Rules:
– A good abductive explanation should provide a plausible cause or hidden event that makes the transition from Observation 1 to Observation 2 reasonable.
– Prefer explanations that reflect everyday commonsense and real-world causal relations.
– Prefer explanations that clearly "bridge the gap" between the two observations.
– Avoid explanations that contradict either observation or introduce unlikely events.
– If both explanations are equally good or equally flawed, choose "tie".
Now, please output your choice ("A", "B", or "tie").

**ROCStories Evaluation Prompt:**

Given the beginning of a short story, we ask models to generate a plausible and coherent continuation.
Story Prompt: "{$input}"
Model A Ending: "{$candidate_A}"
Model B Ending: "{$candidate_B}"

Your Task: Choose which ending provides a more coherent and plausible completion of the story based on the criteria.
Evaluation Criteria:
– Relevance: Incorporate relevant details from the prompt.
– Coherence: Events follow clear causal and temporal progression.
– Clarity: Easily understandable with proper grammar.
– Commonsense Knowledge: Demonstrate correct real-world knowledge.
– Creativity: Provide a fresh and interesting perspective.
– Length: Adhere to specified length requirements.
If both continuations are equally good or equally flawed, choose "tie".
Now, output your choice ("A", "B", or "tie").

Figure 3: Evaluation prompts used by GPT-4o to judge model generations in pairwise comparison. Each prompt defines task-specific criteria for selecting the better output between the model output and Human reference.

**Synthetic Data Generation Prompt (2-Seed Concept Expansion):**

Instruction: You are an expert in commonsense knowledge.
Given a seed of two keywords, complete the set by adding EXACTLY ##num_to_add## keywords.
Your goal is to produce a concept set that can naturally form ONE plausible daily-life event or scene,
which can be expressed in ONE single sentence under 22 words.
Rules for ADDED keywords:

- Each keyword must be a common noun or verb in dictionary form.
- DO NOT use prepositions, articles, or pronouns.
- ADDED keywords must form a coherent *action chain* with the seed keywords.
- The combined keywords (seed + added) MUST be usable to construct one daily-life scene or action.
- Avoid overly abstract or static concepts.
- Output ONLY the added keywords, separated by commas.
- You MUST NOT output any keyword that appears in the seed, even if it seems logically appropriate.

Here are some examples of varied lengths:
—
Example 1:
Seed: passenger, train
Output: station, run
—
...
Your Task:
Seed: #seed_keywords#
Output:

Figure 4: Prompt template used to expand 2-seed concept sets during synthetic data generation. This instruction guides an LLM to add contextually relevant keywords that enable the construction of a single coherent, everyday scenario.

**Prompt for Few-Shot and Multi-Sentence Generation ($D_{\text{dyn}}$, $D_{\text{ms}}$):**

You generate fluent and commonsense-bearing English sentences under strict formatting constraints. Do not add explanations or numbering. Output only the requested sentences.

Instruction: Given ##num_to_add## keywords, generate exactly ##N## commonsense-bearing and diverse English sentences.
Each sentence MUST contain ALL the required keywords (inflectional variants are allowed, e.g., stand→stood/stands).
Requirements:

- The sentence must describe a logically possible everyday situation.
- It must contain **ALL** the provided keywords (inflectional variants allowed).
- Keep the sentence concise ($\leq$ 22 words).
- Separate sentences with a single TAB character (\t).
- Do not add explanations, numbering, or commentary.
- Output exactly ##N## sentence(s).

Diversity rule:

- Vary subject, perspective (first/third person), tone, or setting to ensure distinct expressions.
- Use natural variation while maintaining plausibility.

Keywords: ##problem##
Use the few-shot examples below as inspiration for style and structure, **not as templates to copy**.

**Prompt for Chain-of-Thought Guided Generation ($D_{\text{cot}}$):**

You must follow formatting constraints. Do not add numbering or commentary outside the required reasoning and final sentence. Output both a reasoning step and the final sentence for each example.

Instruction: Given ##num_to_add## keywords, generate exactly ONE commonsense-bearing English sentence based on your lifestyle and background.
Each sentence MUST contain ALL the required keywords (inflectional variants are allowed, e.g., stand→stood/stands).
Requirements:

1. First write a reasoning or description to explain the underlying commonsense connection of the keywords. Start with **"Let's think step by step:"**
2. Then, on the next line, output ONE realistic English sentence that contains ALL the keywords.
3. Keep the sentence concise ($\leq$ 22 words).

Formatting constraints:

- Use exactly **one reasoning paragraph (**$\geq$ **4 sentences)** and **one sentence** per generation.
- Separate each generation with a blank line.
- Do not add numbering, commentary, or bullet points.

Keywords: ##problem##

Figure 5: Prompts used to generate synthetic sentences from expanded concept sets. The top prompt is shared by both dynamic few-shot ($D_{\text{dyn}}$) and multi-sentence few-shot ($D_{\text{ms}}$) strategies; the bottom prompt enforces explicit reasoning via chain-of-thought ($D_{\text{cot}}$). All outputs are constrained to $\leq$ 22 words and filtered for full keyword coverage.

**Quality Scoring Prompt for Candidate Sentences (Per Concept Set):**

You are an expert evaluator of sentence quality and commonsense reasoning.
I will give you a set of concepts, and {num_sentences} candidate sentences generated using different prompting strategies.
Your task:

1. Score each sentence independently from 1 to 10.
2. Higher score = higher quality and commonsense correctness.
3. All scores must be integers.
4. Use the full range (1–10) with these guidelines:
    - **1–3 (poor):** Incorrect, implausible, ungrammatical, or fails to use the concepts meaningfully.
    - **4–6 (average):** Mostly correct but may have minor issues in clarity, grammar, or concept integration.
    - **7–8 (good):** Clear, fluent, plausible sentences that use the concepts well.
    - **9–10 (excellent):** Exceptional clarity, realism, and full concept integration. No errors.
5. If a sentence is marked as “[EMPTY]”, assign it a score of 1.

Evaluate based on:

- **Commonsense correctness**: Is the event plausible and realistic?
- **Concept coverage**: Are the concepts used meaningfully together?
- **Clarity and grammar**: Is the sentence well-formed?

Concept set:
{concept_set}
Candidate sentences:
{sentence_list}

Figure 6: Prompt used by the quality scorer (Gemini-2.5-flash) to assign plausibility scores $Q(y) \in [1, 10]$ to each candidate sentence given a concept set $\mathcal{C}$. Scores guide filtering ($Q(y) < 4$ discarded) and subsequent local diversity selection.

**CommonGen Evaluation Prompt (Main Task):**

Given several concepts (i.e., nouns or verbs), we ask models to write a short and simple sentence that contains all the required words.
The sentence should describe a common scene in daily life, and the concepts should be used in a natural way.

Concepts: {$input}
Model A: {$candidate_A}
Model B: {$candidate_B}

Your Task: Your task is to choose a better sentence from the two candidates. Decide which model’s sentence is better in terms of the naturalness and commonness of the scenes they describe.

Rules:

- A better sentence should describe a common scene in daily life, and all concepts should be used in a natural way.
- You should prefer sentences that use all given concepts.
- A simpler and shorter sentence is preferred if it describes the same scene as the other sentence.
- If you think both sentences are equally good or bad, please choose "tie".

Now, please output your choice ("A" or "B" or "tie").

Figure 7: Evaluation prompt used to compare model-generated sentences against human references on the CommonGen task. This prompt—adapted from the official `CommonGen-Eval` repository—guides GPT-4o to judge based on naturalness, concept coverage.

| **Model** | **Dataset** | **Quality Metrics (↑)** | | | **Diversity Metrics (↑)** | | | |
|---|---|---|---|---|---|---|---|---|
| | | Win-Tie | Cov. | Overall | S-BLEU3 | S-BLEU4 | Vendi | S-Cos |
| Llama-3.1-8B-Inst | Vanilla | 19.0 | 84,7 | 16.1 | 73.4 | 80.3 | 22.8 | 35.1 |
| | CommonGen | 31.7 | 85.8 | 27.2 | 76.4 | 84.7 | 21.2 | 30.7 |
| | **CommonSyn** | **47.3** | **94.5** | **44.7** | 75.5 | 82.9 | 22.7 | 34.2 |
| Llama-3.2-1B-Inst | Vanilla | 6.0 | 64.3 | 3.86 | 79.5 | 84.8 | 27.2 | 49.7 |
| | CommonGen | 20.9 | 84.0 | 17.56 | 79.2 | 87.0 | 21.2 | 34.1 |
| | **CommonSyn** | **31.1** | **89.6** | **27.87** | 79.2 | 86.1 | 24.2 | 38.3 |
| Llama-3.2-3B-Inst | Vanilla | 7.8 | 65.5 | 5.11 | 72.8 | 78.8 | 24.8 | 41.2 |
| | CommonGen | 26.3 | 86.9 | 22.85 | 77.0 | 84.8 | 21.8 | 32.2 |
| | **CommonSyn** | **42.0** | **92.4** | **38.81** | 76.4 | 83.4 | 23.1 | 35.4 |
| Qwen-2.5-1.5B-Inst | Vanilla | 23.0 | 79.6 | 18.31 | 77.7 | 84.9 | 22.1 | 33.3 |
| | CommonGen | 20.6 | 81.1 | 16.71 | 80.0 | 87.5 | 23.8 | 37.8 |
| | **CommonSyn** | **29.3** | **84.0** | **24.61** | 79.6 | 86.7 | **24.1** | **38.2** |
| Qwen-2.5-3B-Inst | Vanilla | 19.1 | 87.9 | 16.79 | 70.7 | 77.7 | 21.3 | 30.6 |
| | CommonGen | 27.6 | 87.1 | 24.04 | 75.7 | 84.1 | 21.9 | 32.4 |
| | **CommonSyn** | **40.3** | **91.6** | **36.91** | 75.6 | 83.0 | **22.8** | **34.5** |
| Qwen-2.5-7B-Inst | Vanilla | 42.9 | 92.3 | 39.60 | 39.2 | 44.1 | 15.8 | 16.3 |
| | CommonGen | 32.5 | 88.6 | 28.80 | 73.2 | 81.8 | 21.4 | 31.2 |
| | **CommonSyn** | **48.4** | **93.0** | **45.01** | **73.7** | 81.0 | **22.5** | **33.7** |
| Qwen-2.5-14B-Inst | Vanilla | 45.2 | 95.9 | 43.35 | 42.4 | 47.5 | 16.5 | 18.6 |
| | CommonGen | 38.5 | 92.9 | 35.77 | 70.8 | 79.6 | 20.9 | 29.7 |
| | **CommonSyn** | **52.0** | **96.0** | **49.92** | **73.2** | **80.6** | **22.1** | **32.6** |
| Gemma-3-1B-IT | Vanilla | 18.2 | 83.9 | 15.27 | 45.5 | 51.1 | 17.2 | 21.0 |
| | CommonGen | 16.5 | 84.9 | 14.01 | 79.8 | 87.6 | 22.0 | 32.6 |
| | **CommonSyn** | **23.3** | **91.8** | **21.39** | 78.7 | 85.7 | **23.6** | **36.6** |
| Gemma-3-4B-IT | Vanilla | 28.7 | 88.4 | 25.34 | 37.7 | 42.3 | 16.1 | 17.4 |
| | CommonGen | 28.7 | 91.0 | 26.12 | 74.9 | 83.2 | 21.2 | 30.6 |
| | **CommonSyn** | **41.2** | **95.5** | **39.35** | **75.4** | 82.6 | **22.4** | **33.4** |
| Gemma-3-12B-IT | Vanilla | 43.2 | 94.0 | 40.63 | 31.7 | 35.7 | 14.7 | 13.6 |
| | CommonGen | 35.9 | 94.1 | 33.78 | 72.2 | 80.4 | 20.5 | 28.7 |
| | **CommonSyn** | **51.1** | **97.4** | **49.77** | **73.7** | **81.2** | **21.8** | **31.7** |
| Gemma-3-27B-IT | Vanilla | 31.9 | 86.7 | 27.66 | 27.0 | 30.2 | 15.0 | 14.7 |
| | CommonGen | 40.3 | 95.6 | 38.53 | 70.7 | 79.2 | 20.3 | 28.1 |
| | **CommonSyn** | **53.6** | **97.9** | **52.47** | **71.7** | **79.2** | **21.4** | **30.7** |

Table 14: Full experimental results across all 11 evaluated models. For S-BLEU and S-Cos, higher scores indicate greater diversity.