SynthLLM: An LLM-based Scalable Synthetic Data Generation Pipeline for Low-Resource Languages

Solmaz Panahi[1], **John Kelleher**[2], **Vasudevan Nedumpozhimana**[2]
[1]Maynooth University, Ireland, [2]Trinity College Dublin, Ireland
solmaz.panahi.2022@mumail.ie, {john.kelleher, vnedumpo}@tcd.ie

Abstract

Large Language Models (LLMs) have enabled scalable synthetic data generation, yet their effective adaptation to low-resource languages remains underexplored. We introduce an LLM-based generate and annotate paradigm to create synthetic datasets for low-resource NLP classification tasks. The framework employs a smaller model for text generation and a stronger model for automatic annotation. Using Farsi Natural Language Inference (NLI) as a case study, we construct a large-scale synthetic dataset of 100,000 labeled instances. We provide a systematic empirical analysis of annotation quality, label-distribution effects, and training regimes. We compare GPT-4o-mini, Aya-23-35B, and DeBERTa as annotators and examine how annotation variability propagates to downstream performance. Our results show that a warm-up phase with synthetic data consistently outperforms data mixing and reversed ordering. Notably, open-source annotation (Aya-23-35B) achieves comparable downstream performance to the proprietary model (GPT-4o-mini), with significant cost implications for deploying pipelines in low-resource settings. The dataset and code are publicly available at `https://huggingface.co/datasets/Solmazp/text2entail`.

Keywords: Synthetic data, LLM-as-annotator, Low-resource

1. Introduction

The success of Large Language Models (LLMs) in NLP tasks depends heavily on access to large-scale labeled training data—a resource that remains scarce for the majority of the world's languages (Joshi et al., 2020). This scarcity is particularly acute for tasks such as Natural Language Inference (NLI), where data creation and annotation require fine-grained linguistic judgement (Gururangan et al., 2018; Pavlick and Kwiatkowski, 2019). Farsi, spoken by over 110 million people, highlights the scale of this gap: the largest available Farsi NLI dataset contains fewer than 10,000 examples (Amirkhani et al., 2023; Khashabi et al., 2021), compared to 570,000 examples for English SNLI (Bowman et al., 2015). The resulting digital divide limits the applicability of NLP advances to speakers of low-resource languages (Okolo and Tano, 2023; HAI, 2025).

One approach to address data scarcity in low-resource languages is cross-lingual transfer learning, in which multilingual models trained on high-resource languages are adapted to lower-resource settings. Models such as mBERT, XLM-R, mBART, mT5, BLOOM, XGLM (Wu and Dredze, 2020; Conneau et al., 2020; Liu et al., 2020; Xue et al., 2021; BigScience Workshop, 2023; Lin et al., 2022) are well-known examples. Although these models demonstrate reasonable zero-shot cross-lingual capabilities for some low-resource languages, their skewed pretraining data toward high-resource languages often necessitates task-specific fine-tuning to close performance gaps (Lauscher et al., 2020; Ahuja et al., 2022; Qin et al., 2025). This creates a circular dependency: fine-tuning is needed to close the gap, but fine-tuning requires the very labeled data that is unavailable. Even instruction-following multilingual models such as BLOOMZ, mT0, and Aya (Muennighoff et al., 2023; Aryabumi et al., 2024; Singh et al., 2024) inherit this limitation: their development required high-quality multilingual instruction data that is expensive to create for low-resource languages, and their performance remains substantially below their English equivalents (Lai et al., 2023; Abaskohi et al., 2024). One widely adopted approach for data creation is back-translation. Many multilingual datasets, such as XNLI, XQuAD, and MASSIVE (Conneau et al., 2018; Artetxe et al., 2019; FitzGerald et al., 2022) are derived through translations of established English datasets. While these resources provide valuable benchmarks, they often introduce artifacts that severely affect the generalization ability of the models (Artetxe et al., 2020).

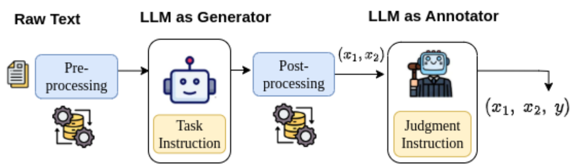

Figure 1: Synthetic data generation pipeline for classification tasks with two textual inputs. An LLM, guided by task instructions, generates input pairs (x_1, x_2) from raw text. A second LLM, prompted with judgment instructions, labels the pairs, producing annotated triplets (x_1, x_2, y) for training.

A more scalable and authentic alternative is to

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 10776–10791
11–16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

generate training data directly in the target language. AI-powered synthetic data generation has emerged as a promising paradigm to address the data scarcity bottleneck (Liu et al., 2024; Long et al., 2024; Nadăş et al., 2025). Within this paradigm, a common approach is to leverage an LLM both as a generator to produce task-relevant text and as a weak annotator. However, the area is underexplored for low-resource languages. Our work is conceptually inspired by TrueTeacher (Gekhman et al., 2023), which demonstrated that annotating model-generated outputs with an LLM teacher produces higher quality synthetic training data than rule-based perturbation of human-written text. However, TrueTeacher was designed specifically for binary factual consistency evaluation in English summarization. We generalize this teacher-based annotation idea into a task-agnostic pipeline for paired-text classification, and instantiate it for three-way NLI in low-resource Farsi. LLMs' natural generation errors yield diverse non-entailed hypotheses without explicit negative-example prompting, reducing annotation cost while preserving label diversity. Our contributions are:

- We propose a cost-effective and task-agnostic framework for synthetic data generation in paired-text classification tasks, and release a 100K Farsi NLI dataset as a concrete output.
- We conduct a systematic comparison of heterogeneous annotator types —proprietary (GPT-4o-mini), open multilingual (Aya-23-35B), and task-specialized (DeBERTa) —and show that open-source annotation achieves comparable downstream performance to the proprietary model.
- We demonstrate that a synthetic $\rightarrow$ gold warm-up regime consistently outperforms data mixing and reversed ordering and identify a saturation point of 10K–20K examples, providing concrete design guidelines for low-resource synthetic data pipelines.

2. Related works

LLMs have increasingly been used to generate and annotate training data, from early instruction-tuning approaches such as Self-Instruct (Wang et al., 2023) and Alpaca (Taori et al., 2023) to broader task-specific dataset construction (Viswanathan et al., 2025; He et al., 2024; Chiang and Lee, 2023; Gunasekar et al., 2023) and annotation (Zheng et al., 2023; Li et al., 2024b). However, ensuring the quality of synthetic data remains a critical challenge. LLMs may produce hallucinations, inconsistencies, or factual errors (Liu et al., 2024; Li et al., 2024a). The reliability of LLMs as annotators is further complicated by sensitivity to prompt variations, biases inherited from pretraining, and substantially degraded performance on low-resource languages (Pavlovic and Poesio, 2024). Therefore, data generated using LLMs can be noisy, especially in low-resource languages where models exhibit degraded performance. Prior work has demonstrated that noisy supervision can improve performance in low-data regimes (Xie et al., 2020; Song et al., 2022; Blum and Mitchell, 1998), but these findings typically assume either random label noise or annotators with high baseline accuracy. It remains unclear whether this generalizes when the annotator itself exhibits degraded performance in the target language. We directly examine this through a systematic empirical comparison of proprietary, open multilingual, and task-specialized annotators, analysing how annotation variability propagates to downstream model performance.

3. Method

Our proposed framework generates synthetic data for any text classification task involving two input texts, such as NLI, summary verification, question-answer verification, translation quality estimation, etc. Figure 1 provides an overview of the proposed pipeline.

First, we use an LLM as a generator to generate pairs of texts D_{gen} where each pair consists of a preceding text (x_p) and a succeeding text (x_s). In the case of the NLI task, x_p will be a premise and x_s will be its candidate hypothesis; in the case of summary verification, x_p will be a document and x_s will be its candidate summary. In our framework, a generator LLM creates candidate pairs by selecting good-quality source texts for (x_p) and generating appropriate successors (x_s) for the task at hand. Then by using an LLM as a judge, each pair in D_{gen} will be annotated with the help of task-specific instructions and generate a labeled dataset D_{lab}, where,

$$D_{lab} = \{(x_p, x_s, y) \mid (x_p, x_s) \sim D_{gen}, y \sim p(y \mid x_p, x_s)\}$$

An annotator LLM generates the label y for a given pair of input texts $x = (x_p, x_s)$ according to the instruction.

This approach offers two key benefits. First, **scalable data generation and labelling**: by leveraging LLMs for both generation and annotation, the pipeline enables efficient creation of large-scale datasets at low cost, which is particularly valuable for low-resource languages where manual annotation is expensive and scarce. Second, **task-aligned negative sampling**: when generated samples are labeled as negative based on a criterion (instruction), they serve as meaningful negative examples that reflect real-world failure modes (e.g., hallucination, irrelevance). These curated negative samples can support contrastive learning or reward-based training by helping models distinguish valid inferences from invalid inferences.

4. Case Study: Data Generation for Farsi NLI

Natural language inference (NLI) is a pairwise input task where, given a premise p and a hypothesis h, the objective is to predict if the premise entails, contradicts, or is neutral towards the hypothesis. Farsi is a low-resource language, and it has limited resources for NLI. One of the NLI datasets for Farsi is the ParsiNLU benchmark dataset (Khashabi et al., 2021), which contains 2.7K manually crafted samples, split into training, validation, and test sets. This data was constructed via two complementary strategies: (i) manually authoring examples from Persian (Farsi) sources with a focus on naturally inferential sentence pairs (e.g., those containing discourse markers like "but"); and (ii) translating and post-editing samples from the English MNLI dataset. The FarsTail dataset (Amirkhani et al., 2023) is another existing resource of 10k examples derived from multiple-choice questions, using a semi-automatic procedure analogous to the SciTail dataset (Khot et al., 2018). While these corpora represent important progress for Farsi NLI, their limited scale constrains their effectiveness for training robust and generalizable inference models.

Using the pipeline described in Section 3, we first generated premise–hypothesis pairs (D_{gen}) and subsequently labeled them with a stronger LLM annotator to obtain the final NLI dataset (D_{lab}).

4.1. Premise and Hypothesis Generation

We begin our generation process of NLI data by selecting high-quality samples as premises. The premises are drawn from the Farsi portion of the XLSum dataset (Hasan et al., 2021), ensuring that they are human-written and exhibit a natural linguistic distribution. Since the input texts in XLSum are significantly longer than the samples in the gold standard ParsiNLU dataset, we apply a filtering process to extract shorter samples with a length distribution similar to that of the gold dataset. Then we generate summaries of selected premises by using an LLM generator as their corresponding hypotheses. Here, our fundamental assumption is that the correct summaries of premise samples will be their entailment hypotheses and incorrect summaries will be their contradictory or neutral hypotheses. For hypothesis generation we use mT5-base model, a multilingual variant of the T5 model (Raffel et al., 2020), fine-tuned on Farsi human-annotated summarization data (Farahani et al., 2021)[1]. This domain adaptation enhances fluency and stylistic alignment with native Farsi text.

While directly prompting a large language model to generate hypotheses for different classes is a straightforward approach, using mT5-base is computationally more efficient and scalable, allowing for cost-effective generation of large quantities of data. We deliberately avoided selecting the best-performing summarization model: its natural generation errors serve as a valuable source of contradictory and neutral hypotheses. Conversely, too weak a generator produces incoherent hypotheses that the annotator LLM cannot reliably label, introducing noise that degrades overall dataset quality. Our choice, therefore, reflects a trade-off: the model is sufficiently strong to produce fluent, informative hypotheses, yet imperfect enough to maintain a controlled error rate that yields diverse non-entailed cases.

To generate summaries, we adopt the standard T5-style prompt formulation, structured as: `"summarize:"{input_text}`. This prefix-based conditioning follows the original T5 design, where tasks are specified via textual prefixes, making the framework readily extensible to other text-to-text tasks [2]. We configure generation with a maximum output of 100 tokens and employ beam search (beam size=2) to reduce variability and mitigate hallucination. However, we observed that over 90% of the generated summaries began with a news agency name (e.g., *ISNA news wrote:*), even when such attributions were absent in the original text, indicating a bias inherited from pretraining on journalistic corpora. To mitigate this issue, we implemented a post-processing step to identify and remove these prefixes while preserving the semantic integrity of the summaries. Figure 2 shows that the generated synthetic data (D_{gen}) exhibits a more pronounced right-skewed distribution for both premises and hypotheses, indicating longer and more variable sentence lengths compared to the ParsiNLU training set. A detailed lexical comparison of synthetic and gold datasets across all dataset sizes, including vocabulary overlap and Type-Token Ratio analysis, is provided in Appendix A.

4.2. Labeled Data Generation

Premise-hypothesis pairs generated in the previous section are labeled as Entailment, Contradiction, or Neutral by using a stronger annotator LLM to generate labeled NLI synthetic data. We

[1] Reported metrics ROUGE-L: 39.96, BERTScore: 79.54, average generation length: 48.72 tokens.

[2] For example, QA can be prompted as `"question:" {question} " context:" {context}`

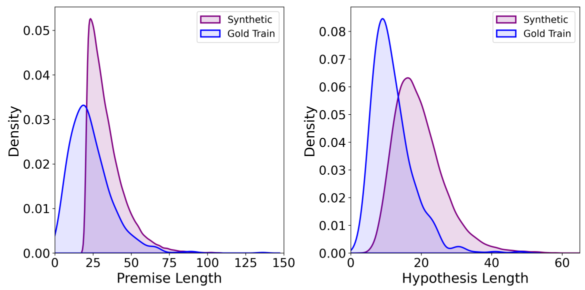

Figure 2: Distribution of sentence lengths for premise (left) and hypothesis (right) in the synthetic and gold training datasets.

chose one of the best-performing general-purpose, cost-effective language models, GPT-4o-mini, as our primary annotator LLM for label generation[3]. However, to assess the robustness of our pipeline across different annotator types, we validated our approach using two additional labelers: Aya-23-35B, an open-weight multilingual LLM, and DeBERTa, a smaller task-specialized NLI model. We use the GPT-4o-mini model with zero-shot prompting and without using any demonstration for the labelling. The model is prompted to evaluate the entailment between the premise and the hypothesis by responding with a categorical judgment. The generation is constrained by choosing only the specified tokens, in this case, the class labels: *Entailment*, *Contradiction*, or *Neutral* [4]. We set the temperature to zero to ensure deterministic outputs, minimising randomness in the model's responses. Additionally, we limit the maximum tokens generated to 5 to enforce concise, single-label answers, ensuring that the model strictly adheres to the required categorical response format without generating unnecessary text. We used English prompts, as they were observed to perform better, confirming the findings of (Dey et al., 2024; Abaskohi et al., 2024) that demonstrate that English prompts perform better:

[3]We include a GPT-4-class annotator for context against public Persian benchmarks (e.g., ParsBench shows accuracy of ≈ 85%); among such models, GPT-4o-mini offers a cost- and latency-efficient option while retaining strong benchmark capability, making it practical for large-scale labeling.

[4]Constraint sampling is a technique to filter the logits vector to keep only the tokens that meet the constraints (Huyen, 2025)

Annotation prompt

System prompt: You are a helpful assistant that evaluates whether a hypothesis can be inferred from a premise. Answer using only one of the following labels: Entailment, Contradiction, or Neutral. Do not explain.
User prompt: Premise: {context} Hypothesis: {summary}

The same prompt was used for Aya-23-35B, enabling a controlled comparison under identical prompting conditions. To assess annotation quality, we applied each annotator to the ParsiNLU train set. GPT-4o-mini achieved 77% accuracy, Aya-23-35B achieved 75%, and DeBERTa-NLI achieved 74%, indicating broadly comparable performance across annotators.

We labeled more than 100K data samples, but the synthetic dataset exhibited severe class imbalance (65,492 Entailment, 32,865 Neutral, 18,331 Contradiction). To prevent majority-class bias in downstream training, we balanced this dataset to a target size of 50K through stratified random downsampling. All experiments reported in this paper use this balanced 50K subset; the full 100K dataset is released publicly to support broader future use. We randomly sampled approximately 16K examples from each class without replacement, using a fixed random seed to ensure reproducibility. We annotated the same premise-hypothesis pairs using Aya-23-35B and DeBERTa, creating three parallel datasets with identical inputs but different label assignments. Table 1 presents the label distributions across all annotators alongside the ParsiNLU benchmark. To mitigate the effect of class imbalance during training, we apply a weighted loss function across all experiments. Detailed analyses of hypothesis fluency, annotator disagreement patterns, and computational cost and pricing considerations are presented in Appendix B, Appendix C, and Appendix D respectively.

	Entailment	**Neutral**	**Contradiction**
GPT-4o-mini	16,665	16,665	16,665
Aya-23-35B	25,799	16,199	7,997
DeBERTa	20,700	18,135	11,160
ParsiNLU (train)	275	241	234
ParsiNLU (validation)	104	87	78
ParsiNLU (test)	608	559	501

Table 1: Label distribution across synthetic and gold datasets. GPT-4o-mini shows the balanced subset after downsampling. Aya-23-35B and DeBERTa retain their natural distributions; class imbalance is addressed during training via weighted loss.

5. Experimental setup

This section describes the models, training procedures, evaluation metrics, and experimental design used to assess the impact of synthetic data on Farsi NLI performance.

We used two mT5-based models (mT5-base and mT5-small) trained on the ParsiNLU training dataset as our baseline NLI models following ParsiNLU benchmarks (Khashabi et al., 2021). While we acknowledge more capable models exist (e.g., reasoning-enhanced LLMs), we deliberately selected smaller models without specialized reasoning capabilities as our baseline models to isolate the impact of synthetic data quality from the influence of advanced model capabilities [5].

mT5 is a text generation model; in its standard formulation, label prediction requires the decoder to generate class names as free text. This creates a risk of invalid outputs outside the predefined label space—we observed instances where the model produced erroneous outputs like "entaildiction", a non-existent word that appeared to be a blend of "entailment" and "contradiction". This behavior revealed limitations in sequence generation for label prediction, especially when target labels such as "entailment", "neutral", or "contradiction" spanned multiple sub-word tokens[6]. To mitigate this, we use mT5 for classification by adding a classification head on top of the encoder. Although this design sacrifices the model's full capacity, it aligns better with the nature of the task and yields more stable predictions[7].

We trained new instances of the baseline architecture from scratch under four regimes: (1) fine-tuning only on the synthetic dataset, (2) pretraining on ParsiNLU gold-standard data followed by fine-tuning on synthetic data, (3) fine-tuning on a merged dataset of synthetic and ParsiNLU gold-standard data, and (4) pretraining on synthetic data followed by fine-tuning on ParsiNLU gold-standard data.

We fine-tuned the baseline models on the ParsiNLU training dataset for a maximum of 10 epochs using a weighted loss function. Hyperparameter configurations are provided in Appendix E.

[5]Exploration of larger instruction-tuned models remains valuable future work, particularly for measuring capability scaling effects and synthetic data utility in low-resource settings.

[6]The challenge might arise from the large vocabulary size of the model (250k tokens), which leads to soft output distributions: the model must select among all vocabulary items rather than only among three valid class labels, increasing the likelihood of near-equal probabilities and misclassifications.

[7]`mT5-small` has 300M parameters, while `mT5-base` has 540M. The encoder-only versions contain 172M and 390M parameters, respectively.

The best checkpoint was selected based on ROC-AUC (One-vs-Rest) on the validation set[8]. To control for variance due to stochastic initialization, we ran training with five different random seeds and reported the average of these five values as the corresponding baseline score. All training regimes used the same hyperparameter settings as the baseline model. Each of these models was evaluated on the same ParsiNLU test dataset used for baseline evaluation. We report the ROC-AUC, accuracy, and weighted-F1 scores for all dataset by training regime model combinations in Table 2.

6. Results and discussion

The impact of synthetic data on downstream performance varies with factors such as the choice of annotator LLM and the size of the synthetic dataset. In this section, we analyze how these factors influence the effectiveness of synthetic data. In Sections 6.1, 6.2 and 6.3 we restrict our analysis to the synthetic data labeled using GPT-4o-mini. This controlled setting allows us to analyze scaling behavior without conflating it with annotator variability. We selected GPT-4o-mini as a strong and cost-efficient LLM annotator, consistent with the design goals of our pipeline, and in Section 6.4 we compare the impact of different LLM annotators.

6.1. Synthetic Data Size

We began with a subset of 5,000 synthetic samples and incrementally added more, generating datasets of varying sizes. We trained and evaluated models using each synthetic dataset. Figure 3 presents the performance of mT5-small (top row) and mT5-base (bottom row) models across three training strategies and increasing synthetic data sizes (from 5k to 50k). We found that the cumulative gain from synthetic data is very high when the dataset size is less than 10K (a steep increase in performance), and after that, the gain is moderate (the performance curve is increasing slowly). Compared to mT5-small, the larger mT5-base model shows a clear performance plateau; however, on mT5-small, the performance is still increasing. This pattern suggests that the benefit derived from synthetic data tends to saturate more quickly for larger models as compared to smaller models. We hypothesize that this saturation reflects two complementary fac-

[8]In multi-class classification, ROC-AUC (One-vs-Rest) computes a separate ROC curve for each class against the rest, providing a threshold-independent and class-sensitive performance measure. This allows us to assess how well models discriminate each class independently, making it particularly suitable for selecting the most generalizable checkpoint across diverse training configurations.

Training Regime	mT5-small			mT5-base		
	ROC	ACC	W-F1	ROC	ACC	W-F1
ParsiNLU (Gold)	0.603 ± 0.031	0.341 ± 0.004	0.287 ± 0.041	0.621 ± 0.077	0.381 ± 0.099	0.353 ± 0.129
ParsiNLU $\rightarrow$ Synthetic	0.741 ± 0.011	0.504 ± 0.017	0.498 ± 0.008	0.782 ± 0.013	0.564 ± 0.014	0.562 ± 0.014
Synthetic (GPT-4o-mini)	0.744 ± 0.006	0.480 ± 0.002	0.496 ± 0.016	0.777 ± 0.020	0.529 ± 0.026	0.519 ± 0.030
Synthetic + ParsiNLU	0.744 ± 0.041	0.500 ± 0.015	0.490 ± 0.019	0.761 ± 0.028	0.548 ± 0.046	0.542 ± 0.049
Synthetic $\rightarrow$ ParsiNLU	$\mathbf{0.800 \pm 0.051}$	$\mathbf{0.564 \pm 0.005}$	$\mathbf{0.564 \pm 0.005}$	$\mathbf{0.812 \pm 0.005}$	$\mathbf{0.611 \pm 0.008}$	$\mathbf{0.612 \pm 0.007}$

Table 2: Performance comparison across training regimes for mT5-small and mT5-base using 50K GPT-4o-mini–annotated synthetic samples. Scores are reported as mean ± 95% CI. Best results per model size are in bold. For reference, GPT-4o-mini zero-shot on the ParsiNLU test set achieves 0.75 accuracy and 0.74 macro-F1, representing an upper-bound estimate.

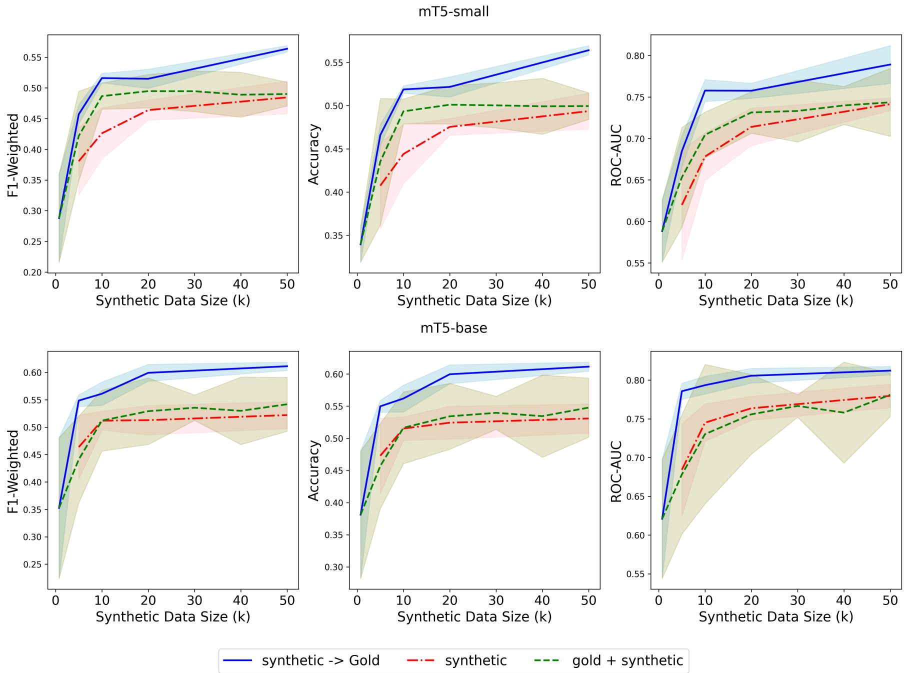

Figure 3: Comparison of mT5-small (upper row) and mT5-base (lower row) across three training strategies and varying synthetic data sizes. Shaded regions indicate 95% confidence intervals across five random seeds. The synthetic $\rightarrow$ gold strategy (blue) consistently outperforms merged and synthetic-only baselines across all metrics and data sizes. Steepest gains occur below 10K samples, after which performance begins to plateau.

tors. First, **annotator noise accumulation**: as dataset size grows, the proportion of label errors introduced by the LLM annotator remains roughly constant, but their absolute count increases. Beyond a certain scale, additional noisy examples may counteract the signal provided by correctly labeled ones, diminishing marginal returns. Second, **model capacity constraints**: mT5-small and mT5-base are relatively compact models whose representational capacity may be saturated after exposure to approximately 10K diverse synthetic examples. This interpretation is supported by the observation that saturation occurs later for mT5-small than mT5-base—a smaller model requires more data to reach its capacity ceiling, while the larger model extracts available signal more efficiently from fewer examples. Together, these factors suggest that simply increasing synthetic data volume yields diminishing returns beyond a threshold; improving annotator quality or model capacity would likely yield greater gains than further data accumulation. This has a practical implication for cost-effective pipeline design: for models of this scale, generating 10K–20K high-quality synthetic examples may be more valuable than generating 50K–100K noisier ones. A lexical analysis presented in Appendix A, shows that gold vocabulary coverage exhibits the same diminishing returns pattern beyond 20K examples, suggesting the performance plateau is partly a lexical phenomenon.

6.2. Effect of Training Regime

From our experimental results in Table 2, we observed that warming up the model on synthetic data, followed by further fine-tuning on gold data

(Synthetic → Gold), achieves the overall best performance on all evaluation metrics. Moreover, Figure 3 shows this two-stage strategy (blue) consistently outperforms the other two (red and green) for all dataset sizes. To complete the training regime comparison, we also evaluated a Gold → Synthetic ordering (ParsiNLU → Synthetic) using GPT-4o-mini labels. This regime consistently underperforms the Synthetic → Gold ordering across both model sizes, confirming that synthetic data is most effective as an initialization signal rather than a fine-tuning target. Based on this observation, Gold → Synthetic experiments were not extended to Aya-23-35B and DeBERTa annotators, as the ordering effect is expected to generalize across annotator types.

The synthetic-only model lags behind other setups across both model sizes, particularly at smaller data scales. This suggests limited generalization when synthetic data is used without any gold supervision. The merged approach is competitive in the mid-range (10K–30K samples), but plateaus earlier than the warm-up strategy. This pattern suggests that combining synthetic and gold data dilutes the value of gold supervision rather than compounding it. Notably, the performance gap between training strategies is more pronounced for mT5-small, highlighting that smaller models benefit more from structured training regimes (e.g., warm-up followed by fine-tuning). These results underscore the importance of training strategy design, especially when operating under low-resource constraints. Beyond accuracy gains, the warm-up strategy also improved optimization stability. We observed more stable gradient norms with lower magnitudes compared to other regimes. This contributed to faster convergence —in many runs, the best ROC-AUC checkpoint was reached within the first few epochs of gold fine-tuning.

6.3. Effect of Balancing Label Distribution

Since the pipeline naturally produces a skewed label distribution, we compared training on the original imbalanced data with class-weighted loss, and the balanced subset. We trained models by using this imbalanced dataset along with the ParsiNLU train dataset and evaluated on the ParsiNLU test dataset with the same training setting as in previous experiments. Figure 4 shows the performance scores using this imbalanced dataset, along with previous performance scores obtained with the balanced dataset. From the results, we observed that models trained on imbalanced synthetic data consistently underperform their balanced counterparts at smaller scales. However, imbalanced training recovers at larger data sizes, suggesting that class imbalance has a diminishing negative impact as overall volume increases. This has practical implications for data curation strategies in low-resource settings: optimizing for balance may yield greater returns than simply scaling volume.

6.4. Effect of Annotator LLM

To evaluate how annotator selection impacts synthetic data quality, we compared three distinct annotation strategies: GPT-4o-mini as a state-of-the-art proprietary model, Aya-23-35B (Aryabumi et al., 2024) as a leading open-source multilingual model with Farsi coverage, and a specialized NLI model based on DeBERTa (Laurer et al., 2022). This selection enables comparison across model types (proprietary vs. open-source), scales (35B vs. smaller specialized), and training objectives (general vs. task-specific). While GPT-4o-mini produced balanced labels (by design), Aya-23-35B and DeBERTa exhibited natural skews toward entailment (62% and 58% entailment rates respectively). These systematic differences likely reflect variations in model architecture, training data distribution, and multilingual calibration. Inter-annotator agreement analysis showed 62% agreement between GPT-4o-mini and Aya-23-35B, and 55% agreement between GPT-4o-mini and DeBERTa, indicating substantial variation in labeling strategies. Figure 5 compares downstream performance across three annotator models (GPT-4o-mini, Aya-23-35B, and DeBERTa-NLI) under different training regimes for both mT5-small and mT5-base, using 50K synthetic samples. All annotators improve over the gold-only baseline when synthetic data is incorporated. Differences between annotators are visible under the synthetic-only and merged regimes. However, under the synthetic→ParsiNLU warm-up regime, performance converges across all three annotators. This suggests that training strategy has a greater impact on downstream performance than annotator choice at this scale.

7. Limitations

Our study has several limitations that constrain the generalizability of findings and suggest important directions for future work. The proposed framework generalizes to any paired-text classification task—including summarization verification, question-answering assessment, and semantic similarity—where annotation distinguishes positive and negative samples. However, our experiment was mainly focused on a single task and language.

We deliberately used smaller models to isolate synthetic data effects, but this choice limits generalizability to modern large language models. Instruction-tuned models with tens of billions of parameters may exhibit different scaling behaviors,

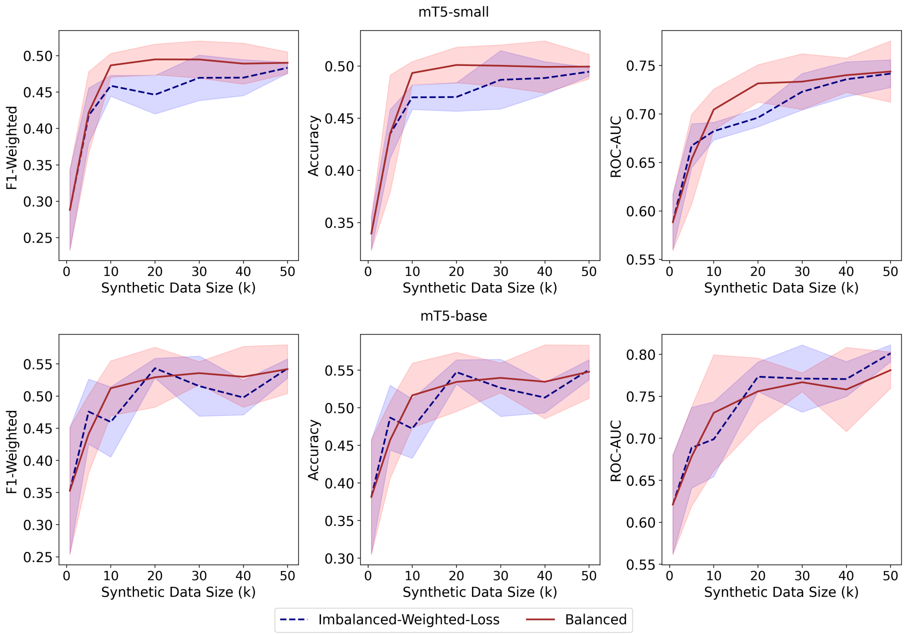

Figure 4: Performance of mT5-small(top row) and mT5-base (bottom row) trained on imbalanced and balanced synthetic datasets across increasing data sizes. Imbalanced training recovers to match balanced performance at higher data volumes.

potentially benefiting from larger synthetic datasets or showing less dependence on training regime optimization. Different architectural families such as encoder-only BERT-style models versus decoder-only GPT-style models may respond differently to synthetic data, and models pretrained on substantial Farsi corpora may require less synthetic augmentation. Future work should use larger models to strengthen claims about the impact of synthetic data on downstream tasks.

Our experiments aimed to evaluate the utility of synthetic data for low-resource languages, leaving a comprehensive exploration of data curation strategies to future work. Regarding coverage, the input data was sourced exclusively from XLSum's BBC News corpus, and this constrains output to journalistic styles. By contrast the gold dataset includes more diverse linguistic styles. For label quality, we relied solely on zero-shot prompting. Future work should investigate whether prompt enrichment strategies —such as explicit world-knowledge exclusion instructions, chain-of-thought reasoning, or few-shot examples —can bring LLM annotation behavior closer to the human standard and improve overall label reliability. Using summarization for hypothesis generation, while computationally efficient, introduces a systematic bias toward Entailment labels. Alternative strategies warrant further investigation, including adversarial generation in which models are prompted to produce contradictory or neutral hypotheses. However, these approaches introduce different trade-offs between computational cost, annotation reliability, and control over label balance.

8. Conclusion

We introduced a scalable synthetic data generation pipeline and demonstrated its effectiveness for Farsi NLI, a language with limited training resources. Our systematic evaluation yields three key insights. First, synthetic data substantially improves performance (+23% accuracy) over gold-only baselines when training regimes are optimized, with a Synthetic $\rightarrow$ Gold warm-up strategy consistently outperforming merged or reversed orderings. Second, marginal gains diminish beyond 10K–20K examples, suggesting that quality and training strategy matter more than raw volume for models of this scale. Third, open-source annotation (Aya-23-35B) matches proprietary quality (GPT-4o-mini) in downstream performance. These findings have practical implications for low-resource NLP. For researchers working with limited budgets and data, our results suggest prioritizing: (1) training regime design over annotator selection, (2) generating 10K–20K diverse examples over 100K noisier ones, and (3) leveraging open-source models for annotation when costs constrain scale. We release our 100K Farsi NLI dataset to support future research. While our experiments focus on a single language and model family, the results highlight broader opportunities for synthetic data generation in low-resource NLP.

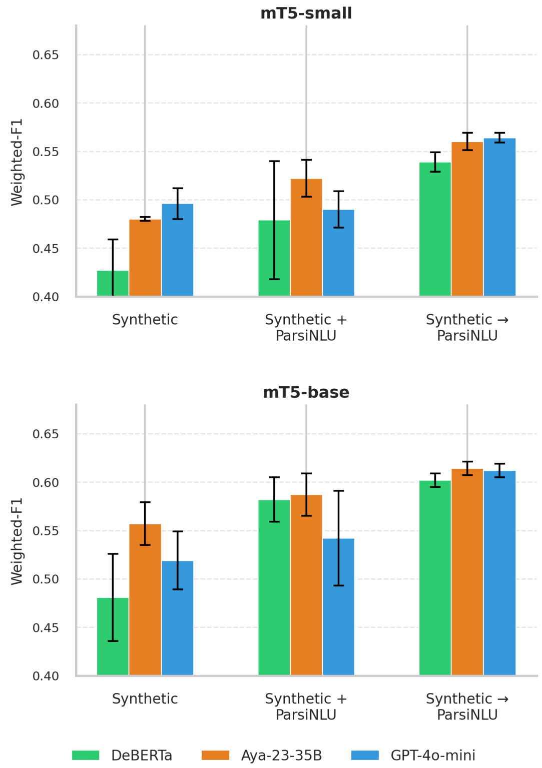

Figure 5: Weighted-F1 comparison across annotators (DeBERTa, Aya-23-35B, GPT-4o-mini) and training regimes for mT5-small (top) and mT5-base (bottom) at 50K synthetic samples. Error bars indicate 95% confidence intervals across five random seeds. Under the Synthetic→ParsiNLU warm-up regime, all three annotators converge, suggesting that training strategy matters more than annotator choice at this setting.

Acknowledgment

This publication has emanated from research supported in part by a grant from Taighde Éireann Research Ireland under Grant number 18/CRT/6049. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

Amirhossein Abaskohi, Sara Baruni, Mostafa Masoudi, Nesa Abbasi, Mohammad Hadi Babalou, Ali Edalat, Sepehr Kamahi, Samin Mahdizadeh Sani, Nikoo Naghavian, Danial Namazifard, Pouya Sadeghi, and Yadollah Yaghoobzadeh. 2024. Benchmarking large language models for Persian: A preliminary study focusing on ChatGPT. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2189–2203, Torino, Italia. ELRA and ICCL.

Kabir Ahuja, Shanu Kumar, Sandipan Dandapat, and Monojit Choudhury. 2022. Multi task learning for zero shot performance prediction of multilingual models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5454–5467, Dublin, Ireland. Association for Computational Linguistics.

Hossein Amirkhani, Mohammad AzariJafari, Soroush Faridan-Jahromi, Zeinab Kouhkan, Zohreh Pourjafari, and Azadeh Amirak. 2023. Farstail: a persian natural language inference dataset. *Soft Computing*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Kelly Marchisio, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. Aya 23: Open weight releases to further multilingual progress.

BigScience Workshop. 2023. BLOOM: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, page 92–100, New York, NY, USA. Association for Computing Machinery.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Krishno Dey, Prerona Tarannum, Md. Arid Hasan, Imran Razzak, and Usman Naseem. 2024. Better to ask in english: Evaluation of large language models on english, low-resource and cross-lingual settings.

Mehrdad Farahani, Mohammad Gharachorloo, and Mohammad Manthouri. 2021. Leveraging parsbert and pretrained mt5 for persian abstractive text summarization. In *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, page 1–6. IEEE.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2022. MASSIVE: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9443–9462, Dublin, Ireland. Association for Computational Linguistics.

Zorik Gekhman, Jonathan Herzig, Roee Aharoni, Chen Elkind, and Idan Szpektor. 2023. TrueTeacher: Learning factual consistency evaluation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2053–2070, Singapore. Association for Computational Linguistics.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Stanford HAI. 2025. Mind the (language) gap: Mapping the challenges of llm development in low-resource language contexts.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024. AnnoLLM: Making large language models to be better crowdsourced annotators. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1487–1506, Mexico City, Mexico. Association for Computational Linguistics.

Chip Huyen. 2025. *AI Engineering*. O'Reilly Media, USA.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Daniel Khashabi, Arman Cohan, Siamak Shakeri, Pedram Hosseini, Pouya Pezeshkpour, Malihe Alikhani, Moin Aminnaseri, Marzieh Bitaab, Faeze Brahman, Sarik Ghazarian, Mozhdeh Gheini, Arman Kabiri, Rabeeh Karimi Mahabadi, Omid Memarrast, Ahmadreza Mosallanezhad, Erfan Noury, Shahab Raji, Mohammad Sadegh Rasooli, Sepideh Sadeghi, Erfan Sadeqi Azer, Niloofar Safi Samghabadi, Mahsa Shafaei, Saber Sheybani, Ali Tazarv, and Yadollah Yaghoobzadeh. 2021. ParsiNLU: A suite of language understanding challenges for persian. *Transactions of the Association for Computational Linguistics*, 9:1147–1162.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Viet Dac Lai, Nghia Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. ChatGPT beyond English: Towards a comprehensive evaluation of large language models in multilingual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13171–13189, Singapore. Association for Computational Linguistics.

Moritz Laurer, Wouter van Atteveldt, Andreu Salleras Casas, and Kasper Welbers. 2022. Less Annotating, More Classifying – Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT - NLI. *Preprint*. Publisher: Open Science Framework.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024a. The dawn after the dark: An empirical study on factuality hallucination in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10879–10899, Bangkok, Thailand. Association for Computational Linguistics.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, Yuxuan Lai, Chongyang Tao, and Shuai Ma. 2024b. Leveraging large language models for nlg evaluation: Advances and challenges.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082, Bangkok, Thailand. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Mihai Nadăş, Laura Dioşan, and Andreea Tomescu. 2025. Synthetic data generation using large language models: Advances in text and code. *IEEE Access*, 13:134615–134633.

Chinasa T Okolo and Marie Tano. 2023. Closing the gap: A call for more inclusive language technologies. *Brookings Institution*.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Maja Pavlovic and Massimo Poesio. 2024. The effectiveness of LLMs as annotators: A comparative overview and empirical analysis of direct representation. In *Proceedings of the 3rd Workshop on Perspectivist Approaches to NLP (NLPerspectives) @ LREC-COLING 2024*, pages 100–110, Torino, Italia. ELRA and ICCL.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S. Yu. 2025. A survey of multilingual large language models. *Patterns*, 6(1):101118.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2023. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(1):1–54.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzemi'nski, Hakimeh Fadaei, Irem Ergun, Ifeoma Okoh, Aisha Alaagib, Oshan Mudannayake, Zaid Alyafeai, Minh Chien Vu, Sebastian Ruder, Surya Guthikonda, Emad A. Alghamdi, Sebastian Gehrmann, Niklas Muennighoff, Max Bartolo, Julia Kreutzer, A. Ustun, Marzieh Fadaee, and Sara Hooker. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. In *Annual Meeting of the Association for Computational Linguistics*.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning with noisy labels: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Vijay Viswanathan, Xiang Yue, Alisa Liu, Yizhong Wang, and Graham Neubig. 2025. Synthetic data in the era of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 5: Tutorial Abstracts)*, pages 11–12, Vienna, Austria. Association for Computational Linguistics.

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.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual bert? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130. Association for Computational Linguistics. Https://doi.org/10.18653/v1/2020.repl4nlp-1.16.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. 2020. Self-training with noisy student improves ImageNet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10687–10698. IEEE.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc.

Appendix A. Lexical Analysis of Synthetic and Gold Data

Table 3 reports vocabulary statistics for the synthetic datasets at each scale alongside the gold dataset. Unique token counts grow steadily with dataset size (22K at 5K examples to 87K at 50K), confirming that additional synthetic examples introduce genuinely new vocabulary. Gold vocabulary coverage—the proportion of gold vocabulary present in the synthetic data—grows from 62.6% at 5K to 79.2% at 50K, but with diminishing returns beyond 20K. This lexical saturation mirrors the performance plateau observed in Figure 3, suggesting that beyond a certain scale, additional synthetic examples contribute less novel vocabulary relevant to the gold distribution. The most frequent tokens in both corpora are identical high-frequency Farsi function words, confirming stylistic consistency between synthetic and gold at the lexical level.

The Type-Token Ratio (TTR) of the synthetic data decreases monotonically with dataset size (0.135 at 5K to 0.053 at 50K), compared to 0.322 for the gold premises. This gap is primarily a corpus-size artifact: with up to 155× more examples, tokens naturally repeat more frequently in larger corpora. To isolate the size effect, we computed TTR on a random size-matched subsample of 749 synthetic premises, obtaining a TTR of 0.27—moderately below the gold TTR of 0.322. This modest gap reflects the synthetic data's reliance on a single journalistic source corpus (XLSum/BBC Farsi), which introduces a degree of stylistic homogeneity compared to the manually curated gold dataset. Despite this, the consistently high gold vocabulary coverage across all scales confirms that the synthetic data remains broadly aligned with the target lexical distribution.

Figure 6 complements this aggregate analysis with per-example vocabulary overlap ratios between premises and hypotheses, each dataset split and the gold test vocabulary, grouped by NLI class. Across all three classes, synthetic distributions broadly match gold train, validation, and test distributions in both shape and median, confirming in-domain lexical consistency. Entailment pairs exhibit higher overlap than neutral and contradiction pairs across all splits, consistent with the semantic proximity inherent to entailing premise-hypothesis pairs. The close alignment between synthetic and gold distributions across classes provides further evidence that the proposed pipeline generates lexically representative data for the target task.

Size	TTR		Unique tokens		Gold coverage	
	Prem.	Hyp.	Prem.	Hyp.	Prem.	Hyp.
5K	0.135	0.102	22,527	9,732	0.626	0.618
10K	0.103	0.073	34,324	13,999	0.686	0.677
20K	0.077	0.052	51,657	20,067	0.738	0.733
30K	0.065	0.043	65,372	24,765	0.764	0.755
50K	0.053	0.034	87,674	32,306	0.792	0.782
Gold (749)	0.322	0.369	5,670	3,274	—	—

Table 3: Lexical statistics across synthetic dataset sizes and gold data. TTR decreases with size as a natural corpus-size effect. Unique tokens grow steadily, confirming vocabulary expansion at each scale. Gold coverage reports the proportion of gold vocabulary present in the synthetic data.

Category	**Fluent**	**Non-Fluent**
Entailment	39670	16190
Neutral	18110	9308
Contradictions	8520	7499
Total	∼ 67k	∼33k

Table 4: Distribution of labels across fluent and non-fluent data

Appendix B. Fluency of Generated Hypotheses

To evaluate the fluency of the generated hypotheses, we implemented a structured annotation protocol focused on native-level Farsi linguistic quality. Through manual evaluation of randomly sampled hypotheses, we identified common error patterns in the generated outputs and incorporated these insights into the prompt refinement process. Each hypothesis was labeled as either "Fluent" or "Not fluent"—without intermediate categories or additional explanations.

Fluency check prompt

System prompt: You are a helpful assistant to evaluate the fluency of text written in Persian (Farsi).
User prompt: For the sentence below:

1. Check if it is grammatically correct, coherent, and sounds natural to a native Persian speaker. You should check for grammar issues, awkward phrasing, broken structure, unclear meaning, hallucinated terms, unnatural repetition, logical errors, etc.
2. Think carefully the meaning of the sentence before labeling.
3. Label it as either: Fluent or Not fluent. Do not explain.
4. Do not fix or rewrite the sentence; only label.

Sentence: {hypothesis}

Table 4 presents the distribution of fluent vs. non-fluent samples across NLI labels. Approximately 67% of generated hypotheses were classified as fluent, indicating reasonable overall quality. However, fluency varies systematically by label: Entailment hypotheses are most fluent (71%), followed by Neutral (66%), with Contradiction hypotheses showing the lowest fluency (53%). This pattern directly reflects our generation strategy. The mT5-base summarization model produces fluent, faithful summaries when functioning correctly—these naturally correspond to Entailment labels. When the

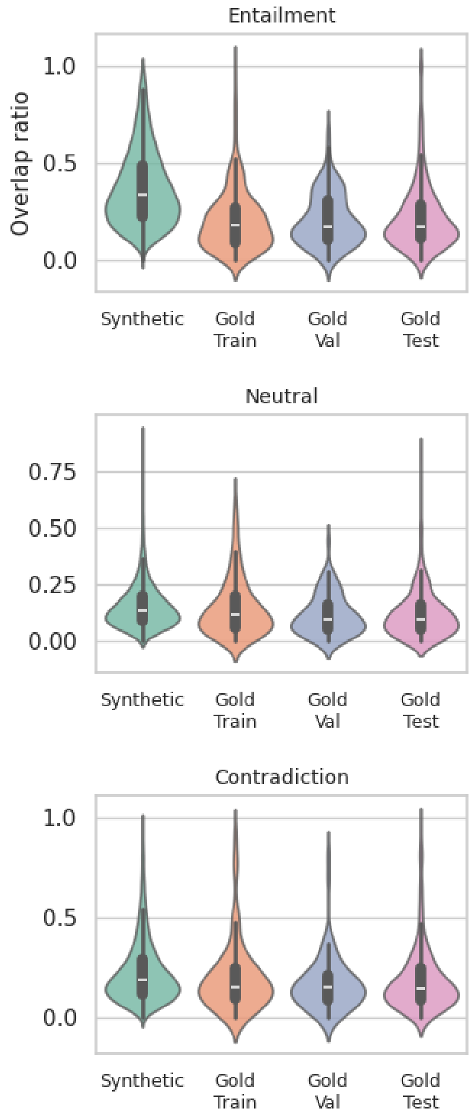

Figure 6: Per-example vocabulary overlap ratio between each dataset split and the gold test set, grouped by NLI class. Overlap ratio is computed as the proportion of tokens in each example shared with the gold test vocabulary. Synthetic distributions broadly match gold train, validation, and test distributions across all three classes, confirming in-domain lexical consistency. Entailment pairs exhibit higher overlap than neutral and contradiction pairs across all splits, reflecting the semantic proximity of entailing premise-hypothesis pairs.

model makes errors (hallucinations, factual mistakes, irrelevant content), outputs are both less fluent and less faithful, yielding Contradiction or Neutral labels. Thus, generation errors serve a dual purpose: they provide negative training examples while simultaneously signaling lower linguistic quality.

Appendix C. Annotation Disagreement Analysis

GPT-4o-mini vs Human Agreement. We evaluate GPT-4o-mini annotation quality by comparing its labels against human annotations on 600 synthetic sentence pairs. The overall agreement between GPT and human labels is 72.6%, with 436 matched cases and 164 mismatches. To guide the human annotation process, two annotators were explicitly instructed to evaluate entailment solely based on the textual content of the premise and hypothesis, without drawing on commonsense reasoning or external world knowledge. Specifically, annotators were asked to apply an "according to" test: given only the premise, does the hypothesis follow? This instruction mirrors the attribution principle formalized in the AIS framework (Rashkin et al., 2023), which defines a valid inference as one a generic reader would affirm when asked "according to the source, does this follow?" –explicitly excluding background knowledge or real-world assumptions beyond the text. Table 5 shows some of the samples that LLM and humans disagree on. Manual inspection of disagreement cases reveals four recurring error patterns:

- **Referential ambiguity:** GPT over-confidently resolves underspecified references, such as ambiguous pronouns (“he”) or unnamed locations (“Isfahan”), leading to unwarranted entailment predictions.
- **World knowledge interference:** The model occasionally draws on external knowledge not present in the premise, producing labels that reflect plausible inferences rather than textual entailment.
- **Entailment fuzziness:** Subtle lexical substitutions blur the boundary between neutral and contradiction, leading to borderline misjudgements.
- **Agency misattribution:** Confusion over speaker or actor identity causes incorrect entailment assignments when the hypothesis shifts attribution relative to the premise.

These disagreements are consistent with challenges reported in human AIS annotation studies, where annotators similarly struggle with referential ambiguity and named entity underspecification despite explicit instructions to avoid world knowledge inference. Note that the human and LLM annotation protocols were not fully aligned: human annotators followed a structured AIS-based framework restricting inferences to textual evidence only, whereas the LLM received a minimal zero-shot prompt without equivalent framing. This asymmetry likely contributes to the observed disagreements.

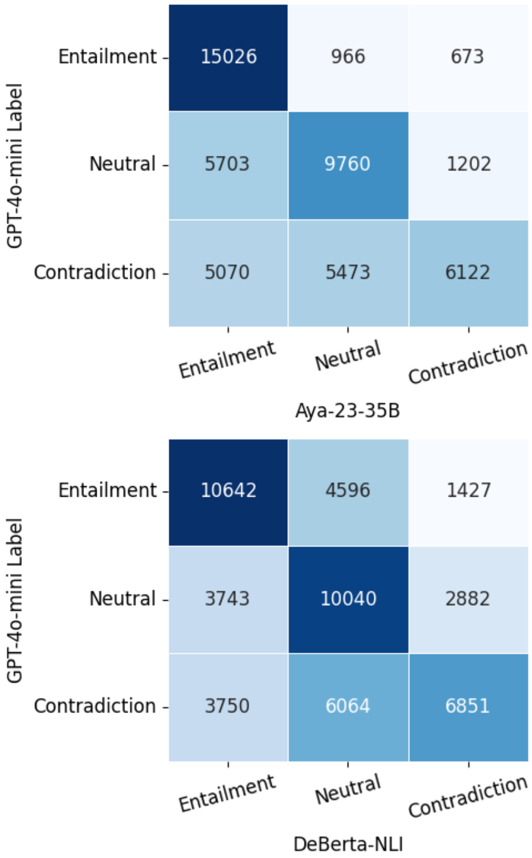

Figure 7: Comparison of pairwise confusion matrices between GPT-4o-mini and other annotators (Aya-23-35B and DeBERTa-NLI) across NLI classes.

Fluency further moderates agreement: for grammatically well-formed premise-hypothesis pairs, agreement rises to 78.9%, whereas degraded or semantically vague pairs reduce agreement to 62.1%. This 16.8 percentage point gap indicates that annotation noise in the synthetic dataset is concentrated in lower-quality generated hypotheses and that fluency-based filtering could be an effective post-processing step to improve label reliability.

GPT-4o-mini vs. Aya-23-35B and DeBERTa. Figure 7 shows pairwise confusion matrices between GPT-4o-mini and the two other annotators across NLI classes. Inter-annotator agreement is 62% between GPT-4o-mini and Aya-23-35B, and 55% between GPT-4o-mini and DeBERTa. Both matrices reveal a systematic pattern: disagreements concentrate on the Contradiction class, where Aya-23-35B and DeBERTa frequently assign Entailment or Neutral to examples GPT-4o-mini labels as Contradiction.

Appendix D. Resources and pricing

We generated over 100,000 hypotheses in approximately 6 hours using a single NVIDIA A100 (80GB) GPU. For synthetic annotation, each labeling request contained approximately 250 tokens. Annotation prompts were submitted in bulk via OpenAI's Batch API, which supports asynchronous processing at a 50% discounted rate (0.15 USD per 1 million tokens)[9]. All requests were completed within 24 hours. Annotation with Aya-23-35B required roughly two hours on two NVIDIA A100 GPUs (without batch processing), whereas DeBERTa completed the same 50,000 samples in about ten minutes on a single RTX 3060 GPU.

Appendix E. Hyperparameters For fine-tuning mT5 models

We used a constant learning rate of $1\mathrm{e}{-4}$ and a batch size of 32 for the training. Optimization was performed using the Adam optimizer with $\beta_1 = 0.9$, and $\beta_2 = 0.999$. For regularization, we applied a dropout rate of 0.3 on the classification head and a weight decay of 0.01. We set encoder/decoder dropout to 0, as preliminary experiments showed a non-zero value of dropout degraded performance. To promote stable training of the model and effective gradient flow, we initialize the linear layers of the classification head using Xavier uniform initialization (Glorot and Bengio, 2010). All bias terms are initialized to zero to avoid random shifts in the activations.

[9] https://openai.com/api/pricing/

Examples
Premise: During the recent economic crisis, he asked the European Commission to take more decisive action, intervene more in controlling member states' economies, and at the same time guide countries toward closer integration. **Hypothesis**: The President of the European Commission demanded greater intervention in controlling member states' economies. **Human:** Neutral **GPT-4o-mini**: Entailment→Reasoning: The premise states that `"he asked the European Commission to take more decisive action" and "intervene more in controlling member states' economies."` This implies that the President of the European Commission is advocating for greater intervention, which aligns with the hypothesis that he demanded greater intervention in controlling member states' economies. Therefore, the hypothesis can be inferred from the premise. **Remark:** There's ambiguity about who `"he"` is, the hypothesis is plausible, but not entailed or contradicted.
Premise: City officials have stated that railway services will gradually return to normal starting early Wednesday morning, and are expected to be operating at an acceptable level by the end of the day. **Hypothesis:** Isfahan railway services will return to normal starting early Wednesday morning. **Human:** Neutral **GPT-4o-mini:** Entailment→Reasoning: The premise mentions that `"railway services will gradually return to normal"` but does not specify which city's railway services are being referred to. While it is possible that Isfahan is included, the information provided does not confirm it, making the hypothesis uncertain. **Remark:** The location-specific detail (Isfahan) is not entailed by the premise.
Premise: Jared Kushner said that around $50 billion would be invested in the region as part of this plan, creating many job opportunities in the West Bank and Gaza. **Hypothesis:** Venezuela's Minister of Energy announced that the country intends to invest $50 billion in the West Bank and Gaza to create many job opportunities. **Human:** Neutral **GPT-4o-mini:** Contradiction→Reasoning: The premise states that Jared Kushner mentioned a $50 billion investment in the West Bank and Gaza, but it does not indicate that this investment is coming from Venezuela's Minister of Energy. The hypothesis claims that Venezuela is making this investment, which contradicts the information provided in the premise. **Remark:** The model treats differences in named entities or specific details as evidence of contradiction, even when those details are not mutually exclusive.
Premise: When they say John Kerry's son-in-law, the U.S. Secretary of State, is of Iranian descent, this cannot be without influence on his view of Iran and his understanding of our country. **Hypothesis:** The U.S. Secretary of State states that when they say he is of Iranian descent, this cannot be without influence on his view of Iran and his understanding of our country. **Human:** Contradiction **GPT-4o-mini:** Entailment→ Reasoning: The premise discusses the influence of the U.S. Secretary of State's Iranian descent on his views, but it does not state that he himself makes this claim. The hypothesis suggests that the Secretary of State explicitly states this influence, which is not confirmed or denied by the premise. Therefore, we cannot definitively conclude that the hypothesis is true or false based on the premise alone. **Remark:** There is misattribution of identity and speaker.

Table 5: Representative disagreement cases between human annotations and GPT-4o-mini. Divergences commonly arise from referential ambiguity, underspecified named entities, and implicit world knowledge assumptions. These examples highlight the intrinsic ambiguity of NLI and the sensitivity of LLM judgments to interpretive framing and reasoning prompts.