

A Monolingual LoRA-Tuned Qwen2.5-3B System for Arabic Context-Based Question Answering

Hasan Alkhder
Sakarya University, Türkiye
hasan.alkhder2@ogr.sakarya.edu.tr

Maria Hanini
Imperial College London
m.hanini@imperial.ac.uk

Imane Hocine
SnT, University of Luxembourg
imane.hocine@uni.lu

Maher Pasa
Altınbaş University
engmaherpasa@gmail.com

Amro Najjar
Luxembourg Institute of Science and Technology
amro.najjar@list.lu

Abstract

In this paper, we describe our submission to the WMT26 Multilingual Instruction Shared Task (MIST), Sub-task 1 (context-based question answering). We fine-tune Qwen2.5-3B-Instruct with a LoRA adapter on 3,112 extractive-QA examples drawn from the answerable subset of TyDi QA. We submit outputs for the monolingual Arabic (question and context both in Arabic) instances of the official qa-context test set. Our system reaches a mean token accuracy of 0.9643 and produces accurate extractive spans. On the held-out split, fine-tuning roughly doubles both exact match (23.81 $\rightarrow$ 53.97) and token-level F1 (37.00 $\rightarrow$ 68.52) relative to the base model. The adapter corrects over-refusal tendency in addition to improving span precision. The base model emitted a no-answer string in 60% of passages with answers, versus 0% for the fine-tuned model. We report error analysis and discuss the model’s weaker behaviour on cross-lingual context instances.

1 Introduction

Extractive question answering is a core component of many Natural Language Understanding systems, and has been driven largely by English resources such as SQuAD (Rajpurkar et al., 2016). Multilingual benchmarks including TyDi QA (Clark et al., 2020), XQuAD (Artetxe et al., 2020) and MLQA (Lewis et al., 2020) have since extended coverage, but performance still degrades for languages with fewer resources and for models small enough to run on a consumer GPU. Arabic is a case in point. Recent surveys identify the scarcity of benchmark

datasets, limited text-processing tools, and a persistent reliance on translations of English datasets as central obstacles for Arabic question answering (Essam et al., 2024). Despite dedicated efforts such as ARCD (Mozannar et al., 2019), DAQAS (Alami et al., 2023), AraBERT (Antoun et al., 2020) and EAD-Bi-LSTM-BERT (Saoudi and Mohsen Gamoudi, 2025), extractive QA resources for Arabic remain limited compared with English.

The 2026 Multilingual Instruction Shared Task (MIST) (Schmidtová et al., 2026)¹ evaluates instruction-following capability across languages and tasks under a strict constraint that the final system’s total parameter count must be under 10B. The organizers propose three sub-tasks:

- *Sub-task 1* — context-based question answering: given a passage and a question, produce a short answer extracted from the passage;
- *Sub-task 2* — summarization from and to any language;
- *Sub-task 3* — open-ended generation.

Note that Sub-task 1 is cross-lingual in practice. In the official test data the passage and the question are frequently in different languages, and the system must still produce a short answer in the language of the question.

Parameter-efficient fine-tuning has made adaptation of large pretrained models feasible under such constraints. LoRA (Hu et al., 2022) trains a small

¹<https://www2.statmt.org/wmt26/multilingual-instruction.html>

number of low-rank update matrices while leaving the base model frozen, and has been shown to match full fine-tuning on many downstream tasks at a fraction of the memory cost.

In this paper, we present the *alkhder* submission to MIST, which focuses on Sub-task 1, in which a system is given a passage and a question and must produce a short verbatim answer span extracted from the passage. We mainly target the Arabic language. Our submission is therefore best understood as a monolingual case study within the shared task as we deliberately scope it to the 25 test instances where question and passage are both Arabic.

Our system fine-tunes Qwen2.5-3B-Instruct (Qwen et al., 2025) with a LoRA adapter on 3,112 extractive-QA examples drawn from the `answerable_tydiqa` subset of the official sample data. The subset spans multiple languages, but every instance is monolingual and no example pairs a passage in one language with a question in another.

We show that a LoRA adapter with 1.90% of the base model’s parameters, trained in under half an hour on RTX 4090, improves Arabic extractive QA. We find that the base model over-refuses on this task, emitting a no-answer string on 60% of the submitted test instances where an answer was present whilst this is not observed with fine-tuning. In addition, we introduce a character-offset-based completion-only loss masking scheme and conduct error analysis to document common failures. A post-submission analysis, machine-translating the context into Arabic before extraction, eliminates this failure mode on a sample of cross-lingual instances. We submit outputs for the monolingual Arabic context-based QA instances in the official test set, and release our code and adapter.²

The rest of the paper is organized as follows: Section 2 presents the training data, Section 3 describes our methods, Section 4 details the experimental setup and describes the test set composition and our submission. Section 5 presents results on the held-out split and error analysis, Section 6 discusses limitations and future work and Section 7 presents our concluding remarks.

2 Data

2.1 Datasets

Table 1 summarises the data used in this work.

²Adapter: <https://huggingface.co/hasankh/qwen2.5-3b-arabic-context-qa-lora>.

Dataset	Use	Instances
<code>answerable_tydiqa</code>	train	3,049
<code>answerable_tydiqa</code>	held-out	63
MIST test (arb, qa-context)	submission	25

Table 1: Data used

wmt26-mist-sample. Training data is drawn from the official `wmt26-mist-sample` release of Schmidtová et al. (2026), which provides input-output pairs in 27 languages. We filter to examples whose source field equals `answerable_tydiqa` which constitutes the answerable subset of TyDi QA, via the `copenlu/answerable_tydiqa` derivation. This yields 3,112 context-question-answer triples. Because TyDi QA itself is multilingual and our filter does not further restrict by language, the resulting training corpus spans multiple languages (we observe English, Arabic, and Russian examples upon inspection). Each instance is nonetheless monolingual combining a passage, a question, and a gold answer that appears as a verbatim span of the passage, all three in the same language. No instance pairs a passage and a question in two different languages.

MIST test set. The test set is released as JSONL with fields `id`, `prompt`, `task` and `question_lang`. The official `qa-context` test set contains 8,640 instances across 24 languages, of which 96.1% are *cross-lingual* i.e., the passage and the question are in different languages, and the system is expected to return the answer span rendered in the question’s language. Of the 360 instances whose question is in Arabic, only 25 have Arabic in both positions. Because our training data contains no cross-lingual example, we scope our submission to these 25 monolingual Arabic instances, where we have direct evidence of reliable model behaviour. Gold answers are not released, so we report test-set behavior via comparison against a base model.

2.2 Data Processing

Each training example was converted into a single ChatML-formatted string of the form:

```
<|im_start|>system
You are an extractive multilingual
question answering system...
<|im_start|>user
Context: {context}
Question: {question}
Answer:
```

```

<|im_end|>
<|im_start|>assistant
{answer}
<|im_end|>

```

The same template is used at inference time for both our fine-tuned model and the base model. We reserved 2% of the training data (63 examples) as a held-out evaluation split, sampled uniformly at random with a fixed seed (42).

We apply a filter to `source==answerable_tydiqa`, which selects the answerable subset of TyDi QA. Unanswerable questions were already excluded by the derivation and the model is not shown an example in which the correct response is to abstain.

The absence of cross-lingual examples is inherent to the released `answerable_tydiqa` derivation. Using an automated language check over all training instances, 96 are flagged as potential mismatches (3.1%), all of which manual inspection confirmed as false positives (very short questions, embedded foreign proper nouns, or closely related Cyrillic-script languages). No training example pairs a context and a question in different languages. We chose not to synthesize such pairs (e.g., by pairing passages and questions across languages), since a verbatim gold answer can only be validated within its own passage, and we scope our submission accordingly.

3 Method

3.1 Base Model and Adapter

We fine-tune Qwen/Qwen2.5-3B-Instruct (Qwen et al., 2025) with a LoRA adapter (Hu et al., 2022) (rank $r = 32$, $\alpha = 64$, dropout = 0.05) targeting all attention and MLP projection matrices (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`). This yields 59.9M trainable parameters, i.e. 1.90% of the base model’s 3.15B total parameters. The shared task’s 10B budget applies to the total parameter count of the final system including the adapter. At deployment our system consists of the 3.09B frozen base parameters plus the 59.9M adapter (≈ 3.15 B in total), well within the shared task’s limit. Training uses BF16 precision throughout. We do not use quantization (bitsandbytes was incompatible with our CUDA 13.0 environment at the time of training) and do not use Unsloth or QLoRA (Dettmers et al., 2023).

Qwen2.5-3B-Instruct was selected as it is multi-lingual and reports competitive Arabic performance for its size. With 3B parameters, it fits in 24GB of VRAM and leaves substantial margin under the 10B budget. We also opted for LoRA over full fine-tuning because the adapter is only 1.90% of the model’s parameters and trains rapidly on a single GPU. It has previously been applied to Arabic adaptation under comparable shared-task constraints (Alkhder and Abboush, 2026).

3.2 Completion-Only Loss Masking

Extractive QA requires the loss to be computed only on the gold answer span and not on the context or question tokens. The `trl` (Von Werra et al., 2020) library no longer provides completion-only masking for raw-text formats, and template-based token matching is unreliable because subword tokenization is context-dependent (Sennrich et al., 2016). An identical string would yield different token IDs depending on the characters preceding it and the search might not align. We therefore implement a character-offset-based masking scheme, illustrated in Figure 1. Each example is tokenized once with character-to-token offset mapping enabled. The literal response template is located as a plain substring of the raw text, giving a character index b for the answer boundary. Every token whose character span starts before b is assigned the ignore index (-100) in the label sequence, and all remaining tokens keep their true label.

Formally, for a sequence of tokens $t_1 \dots t_n$ with character spans (s_i, e_i) :

$$\ell_i = \begin{cases} -100, & \text{if } s_i < b \\ t_i, & \text{otherwise} \end{cases}$$

Unlike token-level template matching, our approach locates the answer boundary in the raw character string and is therefore robust to tokenizer merge behavior, and requires no dependence on collator internals.

3.3 Training Configuration

We train for 3 epochs with per-device batch size 4 and gradient accumulation 8 (effective batch size 32), a cosine learning-rate schedule peaking at 2×10^{-4} with a 3% warmup ratio, gradient clipping at 0.3, a maximum sequence length of 1024 tokens, and gradient checkpointing. Checkpoints are saved every 100 steps (limit 3), with the best

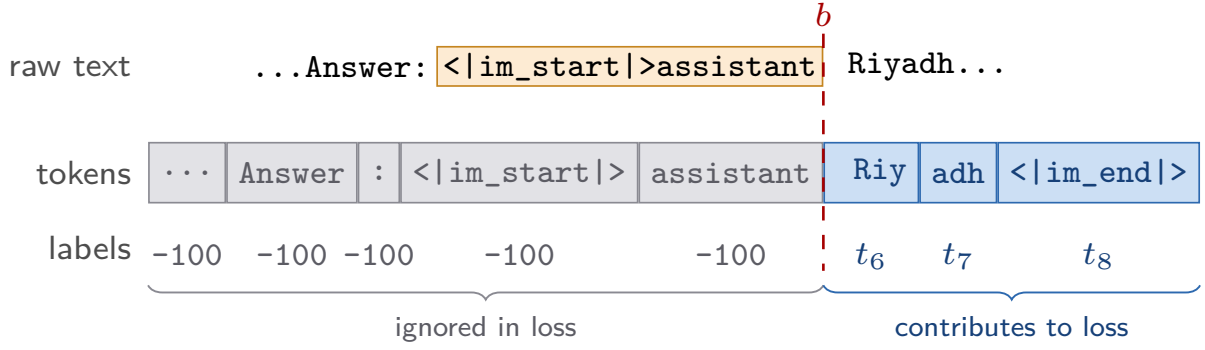


Figure 1: Character-offset loss masking. The response template is located as a substring of the raw text, giving boundary b . Tokens whose spans start before b (grey) receive the ignore index, tokens at or after b (blue) contribute to the loss.

selected by evaluation loss. Training ran on a single NVIDIA RTX 4090 (24GB) and took 20 minutes 58 seconds. Our configuration follows common practice for LoRA fine-tuning models of few-billion-parameter models and was fixed across all experiments. No hyperparameter search over ranks, target modules, or learning rates was performed. Full hyperparameters are given in Appendix A, Table 6.

4 Experiments

4.1 Experimental Setup

Inference. Inference uses greedy decoding with no sampling for deterministic reproducible extractive outputs, with generation stopped at the end-of-sequence or `<|im_end|>` token. We initially set `max_new_tokens=64`, which we found truncated enumeration-style answers mid-word. We thus raised this to 128 for all reported results and for the final submission.

Baseline. We compare against the Qwen2.5-3B-Instruct without our LoRA adapter, executed with the same inference script prompt template and decoding settings above. Only the presence of the adapter differs between the systems.

Metrics. We report exact match (EM) and token-level F1, computed after normalization by stripping Arabic and Latin punctuation and collapsing whitespace. We additionally report mean token accuracy on the held-out split.

4.2 Test Set Composition and Scope

The official test set includes a total of 8,640 qa-context instances across 24 confirmed test languages. Of these, only 340 (3.9%) are monolingual

(question language equals context language) and the remaining 96.1% are cross-lingual making the task mostly cross-lingual. Of the 360 instances whose question is in Arabic, 25 (6.9%) have Arabic in both positions, a share above the median across non-English languages in the test set. The remaining 335 pair an Arabic question with a different answer language.

Given that our training data, while multilingual in aggregate, never contained a cross-lingual example, we scope our primary submission to these 25 monolingual Arabic instances and the remaining test instances receive no output from our system.

5 Results

5.1 Training behaviour

Table 2 summarizes training and evaluation metrics on the held-out split.

Metric	Split	Value
Loss	Train (final)	0.0822
Loss	Eval	0.1658
Mean token accuracy	Eval	0.9643
Entropy	Eval	0.0383
Training runtime	—	20 min 58 s
Trainable parameters (LoRA)	—	59,867,136

Table 2: Training and evaluation metrics on the held-out split.

We observe that fine-tuning converges quickly and cheaply. The adapter reaches a final training loss of 0.0822 and a held-out loss of 0.1658 in 20 minutes 58 seconds. The gap between the two losses is the mild overfitting expected from three epochs over 3,112 examples, and the low output entropy (0.0383) indicates a high confidence about where answer spans begin and end.

Mean token accuracy reaches 0.9643, which might appear to indicate near-perfect extraction. However, this does not necessarily account for the error compounding that occurs during autoregressive decoding. We report EM and F1 below.

5.2 Baseline Comparison

We evaluate the unmodified base model on the identical held-out split using identical prompt construction, decoding parameters and tokenizer. Attaching the adapter doubles EM and F1 metrics on the held-out split (Table 3), improving EM by 30.2 points and F1 by 31.5 points. The fine-tuned model returns a correct span on 34 of the 63 items against 15 for the base model. Since the settings are identical for both runs, the improvement is attributable to fine-tuning.

Model	EM	F1
Baseline (no adapter)	23.81	37.00
LoRA fine-tuned (ours)	53.97	68.52

Table 3: Exact Match and F1 score on the held-out split.

A second effect appears on the test instances. To characterize behavior on the exact instances we submitted, we ran the unmodified baseline on all 25 Arabic qa-context instances under identical decoding settings to the LoRA run, using the training-time system prompt for both. The baseline outputs a no-answer string (لا توجد إجابة) on 15 of 25 instances (60%) where an answer was in fact present in the passage. The fine-tuned model emitted it on none. Fine-tuning thus improves span-extraction precision and corrects refusal tendency in the base model. Table 4 shows example cases.

5.3 Error Analysis

We manually reviewed all 25 submission outputs, together with a sample of cross-lingual qa-context outputs for the Arabic question language. Table 5 summarizes errors we observed.

The truncation mode is resolved by the generation-length fix and does not affect the final submission. For the answer-type mismatch, the model produces an in-passage short span that is nonetheless the wrong one. We suspect this reflects the extractive objective rewarding short high-overlap spans without an explicit answer-type constraint.

5.4 A Translation-Based Approach to the Cross-Lingual Setting

Following the observation that errors we encountered are focused on cross-lingual inputs (Table 5), we evaluated a lightweight translate-then-extract approach post-submission. We sampled 50 cross-lingual qa-context instances with Arabic questions, machine-translated the context into Arabic with NLLB-200, and ran our fine-tuned system unchanged on the translated context, comparing against the same instances without translation.

Without translation, 52% of outputs (26/50) contained foreign-language leftovers or script mixing. With translated context, this dropped to 0% (0/50), with 96% of outputs in clean Arabic, one output (2%) showed a repetition artifact, and no refusals occurred in either condition.

6 Discussion

Overall, the system improves substantially with the adapter, but several limitations remain. The proposed system has no concept of an unanswerable question. The answerable subset we train on contains no example in which the correct response is to abstain, while the system prompt continues to demand abstention when no answer is present. The model has no learned behaviour for abstaining. Fine-tuning did not so much calibrate this behaviour as invert it. Our EM and F1 figures are therefore not comparable to results on the full test set, which includes unanswerable instances.

Moreover, no training example paired a context and a question in different languages. The model performs unreliably on such inputs, which constitute 96.1% of the released qa-context test data. Machine-translating the context into Arabic eliminates the dominant failure mode on a 50-instance sample indicating that the cross-lingual failures stem from context-language mismatch rather than question understanding, and that a translation pass extends our system to the cross-lingual majority of the test set without retraining or additional data collection. All other results are reported on a held-out split of the training distribution, and we do not evaluate on an external benchmark.

Post-submission inspection revealed that the system prompt used during training differs from the submission-generation script. The results reported here use the training-time prompt for both systems and come from a post-hoc re-run, and are therefore not identical to the submitted outputs.

Instance ID	Question	Baseline	LoRA (ours)
qa-context_4_arb_arb	ما هي الفوائد النفسية والاجتماعية... <i>What are the psychological and social benefits ...?</i>	لا توجد إجابة <i>No answer</i>	تعزز احترام التنوع الثقافي والإبداع البشري <i>Promotes respect for cultural diversity and human creativity</i>
qa-context_7_arb_arb	ما هي الأنشطة الأخرى المدرجة تحت القسم ج... <i>What other activities are listed under Section (C) ...?</i>	لا توجد إجابة <i>No answer</i>	فنون وتقاليد أداء العروض <i>Performing arts and performance traditions</i>
qa-context_9_arb_arb	إذا كان تراث مجتمع ما يشتمل على ملاحظات فلكية، فما المجال الذي يغطي هذه الأنشطة؟ <i>If a society's heritage includes astronomical observations, which domain covers these activities?</i>	لا توجد إجابة <i>No answer</i>	الممارسات الاجتماعية والطقوس والاحتفالات <i>Social practices, rituals and festive events</i>

Table 4: Baseline refusal on submitted test instances. English glosses appear in italics below the Arabic text.

Failure mode	Description	Example ID
Truncation	Enumeration-style gold answers were cut off mid-word at max_new_tokens=64. Resolved by raising the generation budget to 128 tokens.	qa-context_8_arb_arb
Unanswerable questions	The model was never trained on unanswerable examples. On unanswerable test questions it fabricates a plausible span instead of a no-answer لا توجد إجابة.	qa-context_15_arb_arb
Answer-type mismatch	The model returns an adjacent span that does not satisfy the question's expected answer type. Asked ما هي الدول التي صاغت قرار الأمن هذا؟ — مجلس الأمن — present in the passage, but does not answer the question requesting a list of states.	qa-context_25_arb_arb
Cross-lingual: verbatim copy	When the passage language differs from the question language, the model sometimes copies a source-language span verbatim instead of rendering it in the question language.	qa-context_2_arb_ces
Cross-lingual: hallucination / script-mixing	On cross-lingual inputs the model occasionally produces text mixing scripts (e.g. Arabic, Bengali and Chinese characters in one span) or topically unrelated to the passage.	qa-context_2_arb_ben

Table 5: Recurring errors observed during manual review.

For future work, the system could benefit from incorporating unanswerable examples into training, restoring calibrated abstention behaviour.

7 Conclusion

In this paper we describe our submission to the WMT26 Multilingual Instruction Shared Task. We present a resource-efficient LoRA fine-tune of Qwen2.5-3B-Instruct for Arabic context-based extractive question answering trained on a consumer GPU in under 21 minutes. Fine-tuning doubles exact match and F1 over the unmodified base model on a held-out split, and eliminates a refusal behavior in which the base model declined to answer despite the answer being available.

References

- Hamza Alami, Abdelkader El Mahdaouy, Abdessamad Benlahbib, Nouredine En-Nahnah, Ismail Berrada, and Said El Alaoui Ouatik. 2023. Daqas: Deep arabic question answering system based on duplicate question detection and machine reading comprehension. *Journal of King Saud University Computer and Information Sciences*, 35(8):101709.
- Hasan Alkhder and Mohammad Abboush. 2026. Sdnlp at amiya 2026: Syrian arabic dialect modeling with lora. In *Proceedings of the 13th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 359–364.
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *Proceedings of the 4th workshop on open-source arabic corpora and processing tools, with a shared task on offensive language detection*, pages 9–15.

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*, pages 4623–4637.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jenimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Mariam Essam, Mohanad A Deif, Hani Attar, Ayat Alrosan, Mohammad A Kanan, and Rania Elgohary. 2024. Decoding queries: an in-depth survey of quality techniques for question analysis in arabic question answering systems. *IEEE Access*, 12:135241–135264.

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

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. Mlqa: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7315–7330.

Hussein Mozannar, Elie Maamary, Karl El Hajal, and Hazem Hajj. 2019. Neural arabic question answering. In *Proceedings of the fourth arabic natural language processing workshop*, pages 108–118.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, and 25 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 2383–2392.

Yassine Saoudi and Mohamed Mohsen Gammoudi. 2025. Ead-bi-lstm-bert: a novel deep learning architecture for arabic question answering systems. *Multi-media Tools and Applications*, 84(38):47289–47311.

Patrícia Schmidtová, Katia Artemova, Seth Aycok, Niyati Bafna, Tom Kocmi, Philipp Koehn, Danni Liu, Nam Luu, Sara Papi, Beatrice Savoldi, Mariya Shmatova, Vilém Zouhar, Maike Züfle, and Pinzhen Chen. 2026. Findings of the WMT26 multilingual instruction shared task: Small models are not yet polyglots.

In *Proceedings of the Eleventh Conference on Machine Translation*. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1715–1725.

Leandro Von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Galouédec. 2020. Trl: Transformer reinforcement learning.

A Hyperparameters

Hyperparameter	Value
<i>Adapter</i>	
LoRA rank (r)	32
LoRA α	64
LoRA dropout	0.05
Target modules	<code>q,k,v,o,gate,up,down</code>
Trainable parameters	59,867,136 (1.90%)
<i>Optimization</i>	
Epochs	3
Per-device batch size	4
Gradient accumulation	8
Effective batch size	32
Learning rate	2×10^{-4}
LR schedule	cosine
Warmup ratio	0.03
Gradient clipping	0.3
Max sequence length	1024
Precision	BF16, no quantization
Gradient checkpointing	enabled
Save steps / limit	100 / 3
Checkpoint selection	best eval loss
Random seed	42
<i>Inference (both systems)</i>	
Decoding	greedy
<code>max_new_tokens</code>	128
Attention implementation	<code>sdpa</code>
<i>Hardware</i>	
GPU	NVIDIA RTX 4090 (24GB)
Training runtime	20 min 58 s

Table 6: Hyper-parameters used to train our model.