RaRaMi at MultiClinSum-2: Cross-Backbone QLoRA Fine-Tuning for Romanian Clinical Case-Report Summarization

BioASQ Lab at CLEF 2026

Radu-George Marin[1], Mihaita Lacusteanu[1] and Rares-Alexandru Iancu[1]

[1]University of Bucharest, Bucharest, Romania

Abstract

This paper describes our submission to the Romanian sub-track of the BioASQ MultiClinSum-2 shared task on multilingual clinical case-report summarization, part of CLEF 2026. We treat the submission as a small empirical study of two factors a resource-constrained team can control when adapting general-purpose multilingual large language models (LLMs) to a new clinical language: the choice of backbone and the amount of training data. Using a single quantized low-rank adaptation (QLoRA) recipe held fixed across all runs, we fine-tune three open-weight instruction-tuned backbones spanning the 1B-8B parameter range (Llama-3.2-1B-Instruct, Qwen-2.5-7B-Instruct, Llama-3.1-8B-Instruct) on the released Romanian corpus and compare them at matched data scales (1,000 and 6,000 records, plus a full-corpus run for the smallest backbone). Two observations organize our results. First, at the 100-record evaluation scale we could afford, the three backbones are statistically indistinguishable on BERTScore F1: the spread across models is smaller than the standard error of the metric. Second, moving from 1,000 to 6,000 training records improves scores for every backbone, though we make no claim that the relationship is linear or that it continues beyond the sizes we tested. Our submitted system, Llama-3.1-8B-Instruct fine-tuned on 6,000 records, reached a test-set BERTScore F1 of 0.86; the organizers' complementary quality dimensions place internal consistency (0.304) far below faithfulness, completeness, and fluency, which we identify as the clearest target for future work. The full pipeline was developed on a single GPU within the constraints of a commercial notebook service, and we release it as a reproducible baseline for clinical summarization in under-resourced languages.

Keywords

clinical summarization, Romanian NLP, low-resource languages, QLoRA, parameter-efficient fine-tuning, multilingual LLMs, BioASQ

1. Introduction

Clinical case reports are a central genre of medical communication: they document a single patient's presentation, investigations, diagnosis, and clinical course in narrative form, and they are the substrate from which practitioners reason about rare presentations, atypical disease courses, and emerging conditions. The volume of published case reports has grown substantially over the past decade, creating a tension between the educational value of the literature and the practical difficulty of identifying clinically relevant material at scale. Automatic summarization of case reports addresses this tension directly, condensing several pages of narrative into a paragraph that preserves the diagnostic reasoning while discarding incidental details.

The BioASQ MultiClinSum-2 shared task [1], part of the BioASQ 2026 challenge [2], targets this problem in a multilingual setting. Organized as part of CLEF 2026, it extends the 2025 MultiClinSum edition to fifteen languages, releasing for each a corpus of full clinical case reports paired with author-written summaries and ranking systems primarily by BERTScore F1, with ROUGE as a secondary lexical measure. The task keeps the corpus- construction and evaluation protocol of the 2025 edition and adds several new languages, including Romanian, which we target here. This paper describes our

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

radu-george.marin@s.unibuc.ro (R. Marin); mihaita.lacusteanu@s.unibuc.ro (M. Lacusteanu); rares-alexandru.iancu@s.unibuc.ro (R. Iancu)

0009-0001-3132-8020 (R. Marin); 0009-0000-5939-2086 (M. Lacusteanu); 0009-0001-5599-2241 (R. Iancu)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

submission to the Romanian sub-track. Romanian is an extended-language sub-track added beyond the four languages covered in the 2025 edition, and it has comparatively limited dedicated resources for clinical NLP. As a Romance language with Latin-derived medical terminology, it shares lexical roots with several better-resourced languages in the task. Prior work on multilingual encoders has shown that such models can transfer across languages without target-language supervision, though the strength of that transfer varies substantially with the target language's representation in pretraining and with its typological distance from higher-resource languages [3, 4, 5]. This motivates, but does not guarantee, reasonable transfer to Romanian. At the same time, Romanian is underrepresented in the training distribution of general-domain LLMs, which limits how much zero-shot competence the model brings to the task before fine-tuning.

We approach the task with parameter-efficient fine-tuning (PEFT) of instruction-tuned multilingual LLMs. Specifically, we adopt a QLoRA recipe over a 4-bit-quantized base model, trained on the Romanian sub-corpus released by the task organizers. The choice of QLoRA reflects a practical commitment: effective participation in a low-resource sub-track should not depend on multi-GPU infrastructure. Our system was developed and trained entirely on a single GPU provisioned through a commercial notebook service, and the pipeline is designed to be reproducible by other teams with similar resources.

We report results across multiple instruction-tuned backbones spanning the 1B-to-8B parameter range, all fine-tuned on the same materialized subset of the released training data to permit fair cross-model comparison. Following the official MultiClinSum evaluation, we report BERTScore F1 against a multilingual encoder, with ROUGE-1, ROUGE-2 and ROUGE-Lsum as secondary metrics. Our best system achieves a development-set BERTScore F1 of 0.8661 on Romanian.

We frame this submission as an empirical study organized around two questions a resource-constrained team can actually decide. The first question is: at a fixed amount of training data, how much does the choice of backbone, model family and parameter count across the 1B-8B range, affect summarization quality on Romanian clinical case reports? The second question is: at a fixed backbone, how does quality change as the amount of training data grows from 1,000 to 6,000 records, and, for the smallest backbone, to the full corpus? To isolate these two factors we hold everything else constant: the prompt, the QLoRA recipe and its hyperparameters, the decoding settings, and the evaluation subset are identical across every run, so the only quantities that vary are backbone identity and training-set size. We do not attempt to identify an optimal system or an optimal recipe; we characterize, descriptively and under a realistic single-GPU budget, how these two controllable factors behave. This framing also delimits what our numbers can support: because the prompt and recipe are fixed, our cross-backbone comparison speaks to backbone choice under this specific configuration rather than in general.

The remainder of this paper is organized as follows. Section 2 situates our approach against prior work on clinical case-report summarization, the MultiClinSum 2025 approaches, and parameter-efficient fine-tuning of multilingual LLMs. Section 3 illustrates the released Romanian corpus and the preprocessing pipeline we applied to filter translation artifacts. Section 4 details the prompt design, model selection, and QLoRA training configuration. Section 5 reports quantitative results on the development set across all backbones, together with qualitative analysis of representative outputs. Section 6 discusses limitations of our approach. Section 7 concludes.

2. Related Work

Clinical case-report summarization. Automatic summarization of biomedical and clinical text has been studied for over a decade, with shared tasks such as MEDIQA [6] establishing standard benchmarks for radiology report and consumer health summarization in English. The MultiClinSum task at BioASQ 2025 [7] extended this line of work to multilingual clinical case reports in English, Spanish, French, and Portuguese, releasing parallel case-summary corpora and using BERTScore F1 as the primary evaluation metric alongside ROUGE [8]. Our submission targets the 2026 Romanian sub-track of the same task family, which keeps the corpus construction and evaluation protocol of the 2025 edition while adding Romanian as a new language.

Approaches in MultiClinSum 2025. Participating systems in the 2025 edition used a wide range of methods. ExtraSum [9] used a fully extractive pipeline across the four official languages, providing a strong reference point for what is achievable without generating new text. MaLei [10] explored iterative self-prompting with frozen instruction-tuned LLMs, showing that careful prompt design can sometimes replace task-specific fine-tuning when compute is limited. Rusnachenko et al. [11] studied decoder-based knowledge distillation from larger to smaller multilingual LLMs as a way to keep summarization quality at a lower inference cost. The closest work to our system is MedGemma-Sum-Pt [12], which fine-tunes a lightweight medically pre-trained backbone for Portuguese clinical summarization on a single GPU. Our work differs in two ways: we target Romanian, an extended-language track without a dedicated medical pre-trained model, and we compare general-purpose multilingual backbones at 1B-8B scale instead of committing to a single model.

Parameter-efficient fine-tuning of multilingual LLMs. Our training recipe builds on parameter-efficient fine-tuning methods that make LLM adaptation feasible on a single GPU. LoRA [13] adds low-rank update matrices on top of the frozen pre-trained weights, while QLoRA [14] further reduces memory by combining 4-bit NF4 quantization of the base model with LoRA adapters in higher precision. We follow the standard QLoRA recipe and use FlashAttention-2 [15] to speed up training and inference. As backbones we use openly released instruction-tuned models from the Llama 3 and Qwen 2.5 [16] families; Qwen 2.5 is particularly relevant for us because it officially lists Romanian among its supported languages. For evaluation we follow the MultiClinSum protocol and report BERTScore F1 [17] as the primary metric, with ROUGE-1, ROUGE-2 and ROUGE-Lsum as secondary lexical-overlap measures.

3. Data and preprocessing

The MultiClinSum-2 organizers released a Romanian training pool of 26,579 case-summary pairs (53,158 paired .txt files), which we cleaned before use. Filtering proceeded in a fixed order: pairs with an empty case or summary were removed; hard length bounds were applied (cases kept to 200-8,000 words, summaries to 30-800 words); the compression ratio (summary length / case length) was constrained to [0.01, 0.95]; pairs whose case and summary were byte-identical were dropped; automatic language identification (langdetect, seeded) was used to remove pairs whose case or summary was confidently non-Romanian, leaving records of uncertain language in place; 3xIQR statistical outliers on log(case words) and on summary length were discarded; and finally exact-duplicate cases and exact- duplicate summaries were removed by SHA-1 hashing. In practice this pass removed 1,039 pairs (3.9% of the pool), all of them at the minimum-length step, 684 cases shorter than 200 words and 355 summaries shorter than 30 words, while the compression, language, outlier, and deduplication checks each fired on zero records, indicating that the released corpus is clean apart from a small number of truncated files. The remaining 25,540 pairs were split 95/5 into 24,263 training and 1,277 development pairs, stratified by case-length quartile so that the development split mirrors the training split in length distribution.

The subsets are constructed up front by: sorting the cleaned training pool by record identifier to eliminate dependence on filesystem order, bucketing records into case-length quartiles, shuffling each bucket with a fixed seed, and interleaving from the buckets in round-robin order to produce a length-balanced sequence. The 1k subset is defined as the first 1,000 records of this sequence, the 6k subset as the first 6,000, and the full subset as the entire sequence.

This construction has three properties we rely on for cross-backbone comparability. First, the subsets are strictly nested: the 1k records are a prefix of the 6k records, which are a prefix of the full set. A 1k experiment therefore makes statements that generalize to 6k without distribution shift, and the same holds for 6k against the full set. Second, the procedure is deterministic across sessions, library versions, and filesystem orderings, because all sources of nondeterminism are eliminated up front. Third, every prefix preserves the length distribution of the full set, so evaluation numbers remain comparable across subset sizes.

Subset hashes (SHA-1) are recorded in a manifest file alongside the data. Each training run verifies

the hash of its input subset against the manifest before training begins, so that any modification of a subset between runs causes training to halt rather than silently use different data. The cross-backbone comparison results in Section 5 are therefore guaranteed to be on identical training inputs, in identical order, regardless of which session each model was trained in.

After preprocessing, the working corpus consists of 24,263 cleaned training pairs and 1,277 development pairs (25,540 pairs in total after filtering). The materialized 1k, 6k, and full subsets are the data sources used throughout the rest of this paper. All experiments reported in Section 5 draw from these subsets exactly as described above.

Our compute budget shaped two important methodological choices that we make explicit here, since they propagate into how the results section should be read.

Firstly, across the three backbones we evaluated (Llama-3.1-8B, Llama-3.2-1B, Qwen-2.5-7B) we trained on the 1k and 6k subsets for all three, and on the full subset for Llama-3.2-1B only. Training the larger backbones on the full corpus was not feasible within our session limits. We treat this as a deliberate scaling design rather than a missing cell in the comparison: the 1k-vs-6k comparison isolates the effect of training-data scale at each backbone, while the per-subset comparison across backbones isolates the effect of backbone choice at each scale.

Secondly, development-set inference is the most expensive operation in our pipeline: for an N-billion-parameter backbone with greedy decoding and prompts of 1,500 tokens, a single development record takes 10–35 seconds depending on N. Evaluating the full 1,277-record development set on every adapter would require significantly more compute than we had available. We therefore adopt a fixed 100-record development subset (the first 100 records of the development split, in deterministic order) and evaluate every adapter on the same 100 records under identical generation hyperparameters. This trades absolute statistical precision (BERTScore F1 on 100 records has roughly ±0.01 standard error compared to ±0.003 on 1,277) for the ability to compare every adapter under matched conditions. We discuss the implications of this choice in Section 5.

4. Method

4.1. Backbone selection

We evaluated three instruction-tuned multilingual LLMs, chosen to span a useful range of parameter counts and to include both major open-weight model families with documented multilingual training data, Llama-3.2-1B-Instruct and Llama-3.1-8B-Instruct from Meta's Llama 3 family [18], and Qwen-2.5-7B-Instruct from Alibaba's Qwen 2.5 family [16].

All three backbones are released as instruction-tuned variants; we do not use any of the base (pre-instruction tuning) checkpoints. This is a deliberate choice: instruction-tuned base models follow chat-style prompt templates out of the box, which the fine-tuning recipe (Section 4.3) relies on for completion-only loss masking.

4.2. Prompt template

All training and inference runs use a single fixed prompt template, expressed natively in Romanian. The prompt is intended to anchor the model in the target language and to make explicit the structural expectations of a clinical case-report summary. Its content requirements track the categories used by the MultiClinSum task organizers to grade summary quality.

The system message specifies the model's role and register (specialist physician writing for medical journals; third-person, past tense, formal medical register), and the task (a single coherent paragraph, no lists or headings). It then enumerates, in order, the content categories the summary should cover when present in the source text — patient demographics, clinical presentation, relevant investigations and key findings, established diagnosis, intervention or treatment, and follow-up evolution — followed by a small number of fidelity rules: use only source-supplied information, omit missing categories silently, and preserve medical terminology in its original form rather than translating drug or procedure names.

Listing 1: System prompt (Romanian). The four blocks specify role and register, task, content requirements, and fidelity rules.

```
Ești un medic specialist care redactează rezumate de cazuri clinice
pentru reviste medicale. Scrii în limba română, la persoana a treia,
la timpul trecut, într-un registru medical formal.

Sarcina ta: să rezumi un raport complet de caz clinic într-un singur
paragraf coerent, fără liste, titluri sau enumerări.

Rezumatul trebuie să includă, în această ordine, atunci când
informațiile sunt prezente în textul-sursă:
1. Datele demografice ale pacientului (vârstă, sex, comorbidități
   relevante).
2. Prezentarea clinică (motivul prezentării, simptome, durata).
3. Investigațiile relevante (examen clinic, analize de laborator,
   imagistică) și constatările lor cheie.
4. Diagnosticul stabilit.
5. Intervenția sau tratamentul administrat (proceduri chirurgicale,
   medicație cu doze dacă sunt menționate).
6. Evoluția și urmărirea pacientului.

Reguli stricte:
- Folosește exclusiv informațiile prezente în textul-sursă. Nu
  inventa date, valori de laborator, doze sau diagnostice.
- Dacă o categorie de mai sus lipsește din text, omite-o fără a o
  menționa.
- Păstrează termenii medicali în forma originală (latină/română).
  Nu traduce denumirile de medicamente sau procedurile.
```

Listing 2: User-turn template. The case report text is substituted for `{case}`.

```
Te rog să rezumi următorul raport de caz clinic:

{case}

Rezumat clinic:
```

We use the tokenizer's 'apply_chat_template' to render the system, user sequence for each backbone, so that the model sees the prompt in exactly the format it was instruction-tuned on. For backbones whose chat template does not accept a separate system role, we detect this at runtime and concatenate the system content into the user turn, separated by a blank line.

During training, each example is rendered as a complete chat (system, user, assistant), tokenized, and presented to the SFT trainer with a completion-only data collator. The collator masks out the prompt tokens so that the language-modeling loss is computed only over the assistant turn — the gold summary. This means the model is never asked to predict the case text or the instructions, which would waste capacity and dilute the gradient signal.

4.3. Fine-tuning recipe

All three backbones are fine-tuned with the same QLoRA recipe and the same hyperparameters, the only thing that varies across runs is the backbone identity and the training subset size (1k, 6k, or full).

Each backbone is loaded in 4-bit NF4 quantization with double quantization, with bf16 compute dtype, using the bitsandbytes [19] implementation. Low-rank adapters are then attached to every projection

in the attention and MLP blocks (q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj), with rank 16, alpha 32, and dropout 0.05. This adds approximately 0.5–1% of the parameter count to the trainable set, depending on backbone, while the underlying 4-bit weights remain frozen.

Training uses the TRL library's SFTTrainer with a completion-only collator. Hyperparameters across all runs: learning rate 2×10^{-4}, cosine schedule with 3% warmup, paged AdamW 8-bit optimizer, 2 epochs over the chosen subset, per-device batch size 1 with gradient accumulation 16 (effective batch 16), gradient checkpointing enabled, maximum sequence length 2,560 tokens, bf16 mixed-precision throughout. Random seed is fixed at 13 for data shuffling and model initialization.

The maximum sequence length deserves comment. The audit pass in Section 3 showed that case reports in the released corpus span roughly 200 to 8,000 words, with the long tail concentrated in a small fraction of records. We selected 2,560 tokens because it covers the median and 75th percentile of training-record token counts without truncation, while keeping per-step activation memory within the budget that allows all three backbones to train within our session limits. Records that exceed this length are truncated from the right, losing portions of the original case that — based on inspection — tend to fall in the late-discussion or literature-comment sections that summaries do not generally reference.

4.4. Inference setup

After training, each adapter is reloaded for inference by re-instantiating the 4-bit base model and attaching the saved adapter via PEFT's 'PeftModel.from_pretrained'. The base is loaded with the same quantization config as during training. Gradient checkpointing is disabled and the KV cache is enabled, both of which are required for tractable autoregressive generation.

Inference is greedy: number of beams 1, do_sample false, maximum new tokens 256, no_repeat_ngram_size 4, EOS token forced from the tokenizer. We do not use beam search for the comparison experiments reported in Section 5. Beam search at width 5 would multiply the per-record generation cost by approximately $4\times$, a trade we cannot afford while evaluating every adapter under matched conditions.

We wrap the generation call in a bf16 autocast context to ensure that activation tensors remain in bf16 throughout. The same prompt template used during training is used for inference, with 'add_generation_prompt=True' so that the chat template ends with the assistant-turn opening marker and the model continues from there.

4.5. Evaluation

We follow the MultiClinSum-2 evaluation protocol [1, 3] and report two families of metrics. ROUGE [8] measures lexical overlap: ROUGE-1 and ROUGE-2 count overlapping unigrams and bigrams, and ROUGE-Lsum is the summary-level longest-common-subsequence variant computed after sentence splitting. BERTScore [17] measures semantic similarity by embedding candidate and reference tokens with a pretrained encoder, greedily matching them by cosine similarity, and aggregating into precision, recall, and F1. BERTScore F1 is the primary metric; ROUGE is a secondary lexical check.

Because the metric configuration affects how the numbers should be read, we state ours explicitly. The development-set scores in Table 1 were computed by us with the Hugging Face evaluate implementations of ROUGE (rouge1, rouge2, rougeLsum) and BERTScore, the latter using the multilingual encoder XLM-RoBERTa-large (model_type = xlm-roberta-large, lang = ro, without baseline rescaling). Because XLM-RoBERTa-large is a multilingual rather than a Romanian-specific encoder, absolute BERTScore F1 values are high and compressed into a narrow band, so small differences between systems carry little signal, a caution the organizers of the 2025 edition raised for non-English BERTScore in general [3].

We evaluate every adapter on a fixed 100-record subset of the 1,277-record development split (the first 100 records in deterministic order), identical across adapters, under the greedy decoding of Section 4.4. On 100 records BERTScore F1 carries a standard error of roughly +/-0.01, against roughly +/-0.003 on the full split, so any two systems separated by less than about 0.02 development BERTScore F1 should be treated as statistically indistinguishable.

The test-set scores in Table 2 are not computed by us. The official Romanian test set of 1,000 case reports is released without gold summaries, so we could not score our own outputs on it; those numbers, including the four complementary quality dimensions (faithfulness, completeness, fluency, consistency), are produced by the organizers' evaluation pipeline. Development and test BERTScore F1 are therefore computed by different pipelines and are only broadly comparable, which is one reason we do not over-interpret the gap between them.

5. Results

Model	Subset Size	ROUGE-1	ROUGE-2	ROUGE-Lsum	BERTScore F1
Llama-3.2-1B	1k	0.3255	0.0769	0.2042	0.8596
Qwen2.5-7B	1k	0.3046	0.0729	0.1997	0.8633
Llama-3.1-8B	1k	0.3271	0.0786	0.2077	0.8645
Llama-3.2-1B	6k	0.3366	0.0803	0.2080	0.8643
Llama-3.1-8B	6k	0.3362	0.0836	0.2099	0.8661
Qwen2.5-7B	6k	0.3185	0.0830	0.2126	0.8692
Llama-3.2-1B	full	0.3414	0.0848	0.2127	0.8638

Table 1
Validation results on the MultiClinSum-2 shared task using 100 evaluation samples for each configuration.

Model	Subset Size	ROUGE-1	ROUGE-2	ROUGE-Lsum	BERTScore F1
Llama-3.1-8B-6k	test	0.3387	0.1495	0.2411	0.8607

Table 2
Final test results on the MultiClinSum-2 shared task after submission.

Model selection for the final submission. For the final submission, we selected **Llama-3.1-8B with a 6k train subset**. Although Qwen2.5-7B with 6k subset achieved the highest BERTScore F1, and Llama-3.2-1B with full training set obtained the best ROUGE scores, Llama-3.1-8B 6k consistently achieved competitive results across all evaluation metrics. The 0.003 BERTScore F1 advantage of Qwen-2.5-7B falls within the ±0.01 standard error of our 100-record evaluation, and Llama-3.1-8B had stronger ROUGE-1 scores at the same scale. We therefore state our selection as a decision under acknowledged noise rather than a ranking read off the table. Because no two systems here are separated by more than the metric's standard error on 100 records, the numbers alone do not single out a winner; we chose Llama-3.1-8B-6k on three explicit grounds: it is the most balanced system across ROUGE and BERTScore rather than winning one metric at the expense of the other; its 8B parameter budget provides more capacity for the long, information-dense clinical inputs this task involves; and the Llama family has documented strength on long-document summarization [20, 21]. We regard these as the operative reasons for the choice, and do not claim that Llama-3.1-8B-6k is quantitatively superior to Qwen-2.5-7B-6k on this development set. Compared to the 1k setup, the 6k configuration improved both ROUGE and BERTScore values, demonstrating that the model benefits from access to a larger portion of the clinical document.

Recent studies have also shown that Llama-based architectures remain highly competitive for long-document summarization tasks, particularly in terms of ROUGE, BERTScore, faithfulness, and robustness when handling large contextual inputs [20, 21]. Furthermore, recent context-aware and hierarchical summarization approaches based on Llama 3.1 demonstrated improved summarization quality while reducing hallucinations in long-context settings [21]. These findings further motivated our decision to select Llama-3.1-8B for the final submission.

In the context of the MultiClinSum-2 shared task, preserving clinically relevant information while maintaining coherence is critical. The 6k subset size allows the model to capture more distributed medical evidence from long clinical notes without the additional computational overhead and potential noise introduced by full-set processing. Furthermore, the 8B parameter scale provides stronger reasoning and summarization capabilities compared to smaller models, making it a reliable choice for the final system submission.

Results interpretation. The experimental results show a consistent improvement when increasing the training-set size from 1,000 to 6,000 records for every backbone we evaluated. We read this descriptively: more training data helped over the range we tested, likely because clinically relevant details are spread throughout the source documents. We do not claim the relationship is linear or that it would continue past 6,000 records; the full-corpus run of the smallest backbone (Table 1) is included precisely to probe where returns begin to flatten, and its ROUGE gains over the 6k setting are modest.

Llama-3.1-8B with 6k subset achieved a ROUGE-1 score of 0.3362 and a BERTScore F1 of 0.8661, demonstrating both strong lexical overlap and semantic similarity with the reference summaries. At the 6k scale the two mid-to-large backbones split the metrics in an interpretable way. Qwen-2.5-7B obtained the highest BERTScore F1 (0.8692) but lower ROUGE than Llama-3.1-8B (ROUGE-1 0.3185 vs 0.3362), whereas Llama-3.1-8B led on ROUGE-1 while trailing slightly on BERTScore F1 (0.8661). Because BERTScore rewards semantic correspondence and ROUGE rewards surface overlap, this is consistent with Qwen producing summaries closer to the reference in meaning but expressed with more paraphrase and lexical substitution, while Llama reproduces more of the reference's exact wording. Given the +/-0.01 standard error of our 100-record evaluation, however, the 0.003 BERTScore F1 gap between the two is within noise, so this describes tendencies rather than establishing a ranking. On the other hand, Llama-3.2-1B with full training data obtained the highest ROUGE scores overall, but as a significantly smaller model, it may be less robust in handling complex clinical summarization scenarios.

Overall, the results suggest that increasing the amount of training data improved quality over the range we tested, while larger models such as Llama-3.1-8B offer a better trade-off between semantic fidelity, lexical coverage, and robustness. Therefore, Llama-3.1-8B with a 6k subset window was considered the most balanced and reliable configuration for the shared task submission.

Final test-set results. Our submitted Llama-3.1-8B-6k system was evaluated on the official MultiClinSum-2 Romanian test set of 1,000 case reports. The system achieved a BERTScore F1 of 0.86, ROUGE-1 of 0.338, ROUGE-2 of 0.149, and ROUGE-Lsum of 0.241, broadly consistent with our development-set estimates. Beyond the overlap metrics, the organizers scored our submission on four complementary quality dimensions: faithfulness (0.699), completeness (0.677), fluency (0.683), and consistency (0.304). Three cluster in a similar range, indicating that the summaries are largely faithful to the source, reasonably complete, and fluent as Romanian prose. Consistency, at 0.304, is the clear outlier, less than half the value of the other three, and is by a wide margin the most informative signal in our results, more so than any difference among the overlap scores. Because BERTScore and ROUGE both compare a summary to a reference and neither penalizes a summary for contradicting itself, this failure mode is invisible to the metrics we optimized against and surfaces only in the organizers' dedicated dimension. We read the low consistency score as internal contradiction within generated summaries, for example incompatible statements about the same clinical fact, or drift in demographic or temporal details across a single paragraph, and note two features of our setup that plausibly contribute: the 256-token generation cap (Section 4.4), which can truncate a summary mid-argument, and greedy decoding, which offers no mechanism to revise an early commitment. Improving internal consistency, through longer generation budgets, decoding strategies that permit revision, or a consistency-aware training objective, is the single clearest direction for future work that this evaluation identifies.

A second observation reinforces the caution urged in Section 4.5. Our submitted system's ROUGE-2 rose from 0.0836 on the 100-record development subset to 0.1495 on the 1,000-record official test set, and ROUGE-Lsum from 0.2099 to 0.2411, while ROUGE-1 and BERTScore F1 moved only slightly. A

near-doubling of bigram overlap between two samples from the same distribution is hard to attribute to anything but sampling variance, and is concrete evidence that our 100-record development estimates, adopted purely for compute reasons, are noisy at the level of individual ROUGE points. This does not change our qualitative conclusions, but it is a reason not to over-read small development-set differences.

6. Limitations

6.1. Compute budget

The experimental program described in this paper was carried out entirely on a single A100 GPU rented through a commercial notebook service, with hard per-session time limits and a finite pool of compute units. This shaped several methodological choices in ways that limit the conclusions we can draw.

First, we did not train every backbone on every training subset. Llama-3.2-1B was trained on the 1k, 6k, and full subsets; Llama-3.1-8B and Qwen-2.5-7B were trained on the 1k and 6k subsets only. The missing top-right cells of the comparison matrix, 8B-class backbones at full-data scale, would have been the most informative configurations for a leaderboard submission, and we cannot rule out the possibility that one of the larger backbones, given the full 24,263-pair training corpus, would have outperformed our submitted Llama-3.1-8B-6k system. Our scaling argument in Section 5 assumes that the data-scale curve at 1B generalizes qualitatively to 7B and 8B, but we do not verify this.

Second, every adapter was evaluated on a fixed 100-record subset of the 1,277-record development set, rather than on the full development set. This was forced by the cumulative cost of running greedy inference on seven adapters across 1,277 records. At our observed inference rates of 100–150 ms per generated token for the larger backbones, full-development evaluation across the program would have consumed approximately 30 hours of A100 time. The 100 record subset reduces this by an order of magnitude but raises the standard error of every reported BERTScore F1 from approximately ±0.003 to ±0.01. Differences between adapters of less than 0.02 BERTScore F1 should therefore be treated as statistically indistinguishable.

Third, we deliberately fixed the QLoRA recipe rather than sweeping it. The hyperparameters (learning rate 2×10^{-4}, LoRA rank 16, alpha 32, dropout 0.05, 2 epochs, effective batch size 16) were held at values standard in the QLoRA literature for tasks of this scale, and identical across every run. This is what makes the two comparisons in Section 5 clean: with the recipe fixed, differences between runs are attributable to the two factors we set out to study, backbone identity and training-set size, and not to per-run tuning. The cost of that design is that we cannot claim our recipe is optimal for any particular backbone or data scale. A sweep over at least learning rate, LoRA rank, and number of epochs is the natural next step and the most valuable single extension of this work, but it multiplies the training budget by roughly the size of the grid and was outside the single-GPU, session-limited budget under which the study was run. We therefore report a fixed-recipe comparison and leave recipe optimization to future work.

6.2. Backbone coverage

We evaluated three open-weight instruction-tuned multilingual backbones in the 1B–8B parameter range. Several axes of variation are absent from our comparison and would be informative to explore:

- We did not include a medically pretrained or medically continued-pretrained backbone (such as MedGemma or BioMedLM) in our final comparison, despite their documented benefits on English-language clinical NLP tasks. We cannot speak to whether medical-domain pretraining transfers meaningfully to Romanian, where the pretraining-data distribution is itself a confound.
- Larger parameter scales. Our largest backbone, Llama-3.1-8B, is small by current frontier-model standards. The behavior of 14B+ models on this task, with the same QLoRA recipe, remains untested.

- Several community-maintained Romanian-specialized fine-tunes of Llama-2 and Llama-3 base models exist (e.g., the OpenLLM-Ro family). A Romanian-specialized starting point might lift absolute performance further, whether it would change the relative ordering of backbones is an open question.

6.3. Prompt design

Several elements of the template were design choices made on the basis of clinical-domain intuition and prior practice rather than empirical comparison: the decision to enumerate six content slots in a fixed order, the inclusion of explicit fidelity rules ("Folosește exclusiv informațiile prezente în textul-sursă"), the choice to omit an explicit length target, the use of Romanian rather than English for the instruction. Each of these may or may not be optimal. Within our compute budget we treated the prompt as a fixed input to the experiment, which means our cross-backbone comparison isolates the effect of backbone choice given our specific prompt, but does not generalize to the effect of backbone choice in general.

7. Conclusion

We presented a QLoRA-based system for Romanian clinical case-report summarization, developed for the MultiClinSum-2 shared task. Working under a constrained single-GPU compute budget, we built a reproducible pipeline from corpus filtering through model evaluation, with deterministic training subsets that make cross-backbone comparisons fair by construction. We trained three instruction-tuned multilingual backbones: Llama-3.2-1B, Qwen-2.5-7B, and Llama-3.1-8B, on identical training data and prompts, and reported development-set BERTScore F1 across configurations. Our submitted system is Llama-3.1-8B fine-tuned on a 6k subset, chosen on the basis of its development-set performance under matched evaluation conditions. The work is a starting point for Romanian clinical summarization rather than a final word: native Romanian medical text, prompt design beyond a single fixed template, and clinician-validated evaluation are all directions we leave to future work. We hope the materialized subset pipeline and the cross-backbone numbers reported here will be useful to other teams working on under-resourced clinical NLP under realistic compute constraints.

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude (Anthropic) in order to: Drafting content, Abstract drafting, Paraphrase and reword, Improve writing style, Citation management, and Formatting assistance. After using this tool, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] M. Rodríguez-Ortega, E. Rodríguez-López, S. Lima-López, A. Mash, M. Krallinger, Overview of MultiClinSum-2 at CLEF 2026: Data, Evaluation, and Results for Multilingual Clinical Case Report Summarization Across 15 Languages, in: E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, 2026.

[2] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. Rodríguez-Ortega, E. Rodriguez-López, N. Loukachevitch, I. Rozhkov, E. Tutubalina, D. Dimitriadis, V. Patsiou, G. Tsoumakas, G. Giannakoulas, A. Bekiaridou, A. Samaras, G. M. Di Nunzio, N. Ferro, S. Marchesin, M. Martinelli, G. Silvello, G. Paliouras, Overview of BioASQ 2026: The fourteenth BioASQ Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering, volume Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026) of *Lecture Notes in Computer Science*, Springer, 2026.

[3] T. Pires, E. Schlinger, D. Garrette, How multilingual is multilingual BERT?, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), Association for Computational Linguistics, 2019, pp. 4996–5001. URL: https://aclanthology.org/P19-1493. doi:10.18653/v1/P19-1493.

[4] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), Association for Computational Linguistics, 2020, pp. 8440–8451. URL: https://aclanthology.org/2020.acl-main.747. doi:10.18653/v1/2020.acl-main.747.

[5] S. Wu, M. Dredze, Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, 2019, pp. 833–844. URL: https://aclanthology.org/D19-1077. doi:10.18653/v1/D19-1077.

[6] A. B. Abacha, Y. Mrabet, Y. Zhang, C. Shivade, C. Langlotz, D. Demner-Fushman, Overview of the MEDIQA 2021 shared task on summarization in the medical domain, in: Proceedings of the 20th Workshop on Biomedical Language Processing, 2021, pp. 74–85. URL: https://aclanthology.org/2021.bionlp-1.8.pdf. doi:10.18653/v1/2021.bionlp-1.8.

[7] M. Rodríguez-Ortega, E. Rodríguez-López, S. Lima-López, C. Escolano, A. Mash, M. Melero, L. Pratesi, L. Vigil-Gimenez, L. Fernandez-Lopez, E. Farré-Maduell, M. Krallinger, Overview of MultiClinSum task at BioASQ 2025: Evaluation of clinical case summarization strategies for multiple languages: Data, evaluation, resources and results, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, 2025, pp. 19–40. URL: https://ceur-ws.org/Vol-4038/paper_2.pdf.

[8] C. Lin, Rouge: A package for automatic evaluation of summaries. in text summarization branches out, ACL 2004, 2004.

[9] S. Rhazzafe, S. Colreavy-Donnelly, N. S. Nikolov, ExtraSum @ MultiClinSum: Extractive summarization of english, spanish, french and portuguese clinical case reports, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, 2025, pp. 534–543. URL: https://ceur-ws.org/Vol-4038/paper_38.pdf.

[10] L. Ren, Y. M. Ng, L. Han, MaLei at MultiClinSUM: Summarisation of clinical documents using perspective-aware iterative self-prompting with LLMs, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, 2025, pp. 520–533. URL: https://ceur-ws.org/Vol-4038/paper_37.pdf.

[11] N. Rusnachenko, X. Liu, J. Chang, J. J. Zhang, Using decoder-based distillation for enhancing multilingual clinical case report summarization, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, 2025, pp. 544–553. URL: https://ceur-ws.org/Vol-4038/paper_39.pdf.

[12] E. T. R. Schneider, F. H. Schneider, E. C. Paraiso, A. S. B. Jr., R. M. O. Cruz, MedGemma-Sum-Pt: A lightweight model for portuguese clinical summarization, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, 2025, pp. 582–594. URL: https://ceur-ws.org/Vol-4038/paper_42.pdf.

[13] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: International Conference on Learning Representations (ICLR), 2022. URL: https://arxiv.org/pdf/2106.09685.

[14] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, QLoRA: Efficient finetuning of quantized LLMs, in: Advances in Neural Information Processing Systems 36 (NeurIPS 2023), 2023. URL: https://arxiv.org/pdf/2305.14314.

[15] T. Dao, FlashAttention-2: Faster attention with better parallelism and work partitioning, arXiv preprint arXiv:2307.08691, 2023. URL: https://arxiv.org/pdf/2307.08691.

[16] Qwen Team, Qwen2.5 technical report, arXiv preprint arXiv:2412.15115, 2024. URL: https://arxiv.org/pdf/2412.15115.

[17] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: Evaluating text generation with BERT, in: International Conference on Learning Representations (ICLR), 2020. URL: https://arxiv.org/pdf/1904.09675.
[18] A. Grattafiori, et al., The Llama 3 herd of models, arXiv:2407.21783, 2024.
[19] T. Dettmers, et al., LLM.int8(): 8-bit matrix multiplication for transformers at scale, NeurIPS 2022, 2022.
[20] T. Li, H. Chen, F. Yu, Y. Zhang, Hera: Hierarchical retrieval-augmented long document summarization with llama 3.1, arXiv preprint arXiv:2502.00448 (2025).
[21] L. Ou, M. Lapata, Context-aware hierarchical merging for long-document summarization using llama 3.1, Findings of ACL (2025).