Small LLMs for Medical NLP: a Systematic Analysis of Few-Shot, Constraint Decoding, Fine-Tuning and Continual Pre-Training in Italian

Pietro Ferrazzi[1,2], Mattia Franzin[1], Alberto Lavelli[1], Bernardo Magnini[1]
[1]Fondazione Bruno Kessler, Povo, Trento, Italy
[2]University of Padova, Padova, Italy

Abstract

Large Language Models (LLMs) consistently excel in diverse medical Natural Language Processing (NLP) tasks, yet their substantial computational requirements often limit deployment in real-world healthcare settings. In this work, we investigate whether "small" LLMs (around one billion parameters) can effectively perform medical tasks while maintaining competitive accuracy. We evaluate models from three major families—Llama-3, Gemma-3, and Qwen3—across 20 clinical NLP tasks among Named Entity Recognition, Relation Extraction, Case Report Form Filling, Question Answering, and Argument Mining. We systematically compare a range of adaptation strategies, both at inference time (few-shot prompting, constraint decoding) and at training time (supervised fine-tuning, continual pretraining). Fine-tuning emerges as the most effective approach, while the combination of few-shot prompting and constraint decoding offers strong lower-resource alternatives. Our results show that small LLMs can match or even surpass larger baselines, with our best configuration based on Qwen3-1.7B achieving an average score +9.2 points higher than Qwen3-32B. We release a comprehensive collection of all the publicly available Italian medical datasets for NLP tasks, together with our top-performing models. Furthermore, we release an Italian dataset of 126M words from the Emergency Department of an Italian Hospital, and 175M words from various sources that we used for continual pre-training.

Keywords: medical natural language processing, small llms, model adaptation, continual pre-training

1. Introduction

Large Language Models (LLMs) have achieved remarkable performance across a wide range of medical Natural Language Processing (NLP) tasks, from clinical concept extraction to question answering. Recently, attention has turned to the so-called "small" LLMs (SLLMs)—models with around one billion parameters—which have become the focus of intensive research (Wang et al., 2025). This shift raises a new question: can small, resource-efficient LLMs perform medical tasks effectively? The answer carries significant practical implications, as hospitals, clinics, and healthcare organisations often operate under strict computational and financial constraints that make large-scale models impractical to deploy (Crema et al., 2024). To this extent, a plethora of methods has emerged, aiming to enhance effectiveness and adapt SLLMs to specialized objectives and downstream applications, including few-shot prompting, constraint decoding, fine-tuning and continual pre-training. Such approaches attempt to overtake the boundaries that SLLMs' parametric knowledge often exhibits with respect to larger models and the challenges of providing consistent, structured outputs.

In this work, we investigate the state of SLLMs in a range of medical NLP tasks in Italian, systematically assessing the impact of different techniques and identifying the most effective methods. We build our analysis on a curated and comprehensive collection of all publicly available Italian datasets for medical NLP tasks. By systematically evaluating different adaptation strategies, we identify the most effective approaches and ultimately obtain a single compact model that outperforms models up to 30 times larger. To enable the evaluation of continual pre-training, we collect a large dataset of medical text from several sources. Our contribution can be summarized as follows:

- We provide a systematic evaluation of the impact of few-shot prompting, constraint decoding, fine-tuning, and continual pre-training on SLLMs for Italian medical tasks.

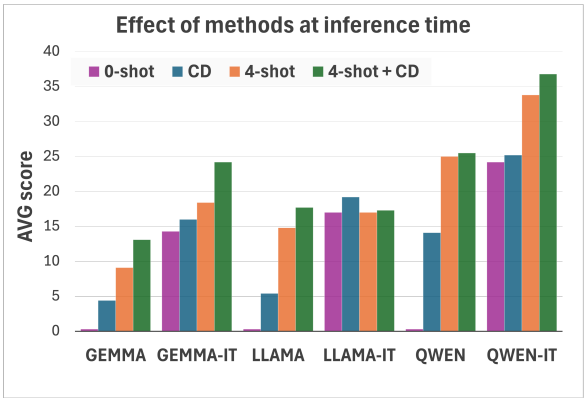
To reach the evaluation goals, we curate and release the following resources[1]:

- the first comprehensive collection of datasets for NLP medical tasks in Italian;
- a new medical dataset in Italian composed of 300 million words from both clinical settings and various sources;
- a Small LLM that outperforms bigger models on medical tasks in Italian.

We designed our publicly available codebase[2] to be easily extensible to new tasks and models, hoping to provide a useful tool to researchers.

[1] huggingface.co/collections/NLP-FBK/small-llms-for-medical-tasks-italian
[2] github.com/ferrazzipietro/llms-for-medical-nlp

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 2440–2457
11–16 May 2026.



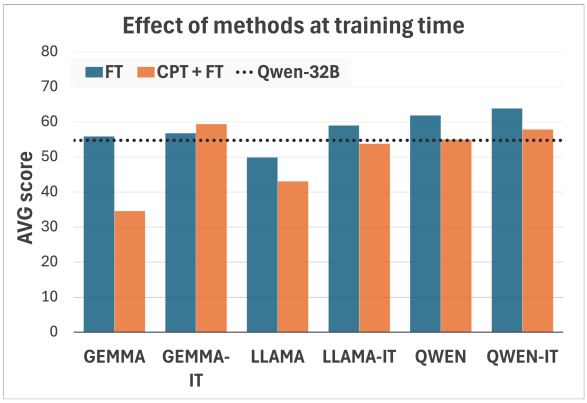


Figure 1: Average performances of 1B LLMs on the 14 medical sub-tasks when different methods are applied at both inference (**left**) and training (**right**) time. Exposing models to Fine-Tuning (*FT*) turns out to be the most effective approach overall, consistently outperforming the baseline (Qwen3-32B with 4-shot). Continual Pre-Training (*CPT*) has a positive impact with respect to simple FT only in one case (gemma-3-1b-it). *4-shot* is consistently better than Constraint Decoding (*CD*), and the combination of the two shows to be beneficial.

2. Methods

In this work, we evaluate how effectively small LLMs can perform on a diverse set of medical tasks, exploring how different adaptation strategies can push their performance to the limit. Our goal is to develop a single, versatile model capable of handling multiple clinical tasks. To achieve such a result, we adopt several strategies that have been proposed to specialise LLMs for domain-specific applications, both at inference and training time. Zero-Shot approaches rely only on models' pre-trained knowledge, while Constraint Decoding (Geng et al., 2023) and Few-Shot learning (Brown et al., 2020) focus on enforcing pre-defined output structures. Supervised Fine-Tuning aligns the model with complex instructions (Zhang et al., 2026), while continual pretraining exposes the model to domain-specific data to improve its medical knowledge (Shi et al., 2026).

Zero-shot prompting We first assess model performance in a zero-shot setting, where the model is only provided with a description of the task and the input text on which to perform it. In this scenario, the model relies entirely on its pre-trained knowledge to generate outputs, without seeing any task-specific examples. Zero-shot evaluation serves as a baseline for understanding how well the model can generalize to medical NLP tasks out of the box.

Constraint decoding Generating structured outputs is critical for real-world systems, which often require formatted outputs to be automatically integrated into downstream components. Correct predictions might turn unusable when presenting formatting errors. Therefore, we employ a constraint decoding method based on *outlines*[3]. Once a valid output schema is defined (e.g., a JSON template), the models' decoding process dynamically masks out all tokens that would lead to an invalid structure at each generation step. For instance, if the model is required to generate a valid JSON string, the decoder ensures that brackets and quotation marks are properly closed and that keys and values follow the expected syntax. This approach restricts the model's generation to comply with the expected format, reducing parsing errors and ensuring consistency (Geng et al., 2023).

Few-shots Another widely used approach to enforce output formatting is few-shot prompting (Brown et al., 2020), where the model is provided with a small number of examples demonstrating the task. These examples help the model better understand the task instructions, improve adherence to the expected output format, and often lead to substantial performance gains, especially for smaller models that may not generalize as well as bigger ones from pre-training alone. We select the examples by random sampling from the training split of each dataset.

Instruction-tuning Fine-tuning the model on a dataset of task instructions paired with expected outputs is an effective strategy to enhance LLM per (Zhang et al., 2026). This approach helps the model learn to follow instructions more reliably, improving both the accuracy of the predictions and the consistency of the generated output structure. Instruction tuning is particularly beneficial when working with small LLMs, as it allows them to leverage prior knowledge while aligning closely with task-specific

[3] https://github.com/dottxt-ai/outlines

expectations. We fine-tuned models on the training split of each dataset.

Continual-pretraining The most resource-intensive strategy to adapt LLMs is continual pretraining, which involves further training the model on domain-specific corpora before handling downstream tasks (Shi et al., 2026). Such a method presents very high data and resource requirements, making it less explored than the others described above. For this reason, we dedicate the next Section to an extensive description of our approach to it.

2.1. Continual Pre-Training

Continual Pre-Training (CPT) consists of further training existing LLMs on next-token prediction over a large corpus of raw textual data, aiming to better inform models on domain-specific knowledge (Shi et al., 2026). In the medical context, it results in exposing models to large volumes of clinical text and biomedical literature, aiming to internalize domain-specific terminology, syntax, and reasoning patterns, resulting in more accurate and contextually appropriate outputs (Luo et al., 2022; Chen et al., 2023). This requires significant computational resources and carefully curated data.
We applied this method to provide models with a comprehensive foundation of specific linguistic and terminological patterns of Italian medical text.

CPT data We used two complementary datasets (Table 1) specifically designed for domain adaptation in the Italian medical and emergency care context. First, we collected a dataset of around 278M words (the **scientific dataset**), combining the data presented by García-Ferrero et al. (2024) with two new sources. The former is composed of multiple high-quality Italian scientific sources, including Med CommonCrawl, drug instructions, medical Wikipedia, the E3C corpus, and others. The latter is built on two sources, namely the medicine-related thesis from the University of Padova[4], and several editions of Pensiero Scientifico and Zadig (Italian scientific publications). We obtain the thesis by scraping the web page, according to the requirements of their the CC0 licence[5]. We recieved Pensiero Scientifico and Zadig volumes from the editors, with a restrictive license which allows training of automatic systems but do not consent to publication of the data.
Then, we create a **clinical dataset** consisting of around 126M words from 1,972,254 anonymized Electronic Health Records, provided by the Fenice Network [6] in collaboration with the Mario Negri Institute and San Giovanni Bosco hospital. All personal identifiers have been removed or replaced (e.g., patient names are substituted with `NOME_PERSONA` and mobile phone numbers with `NUM_TELEFONO`) to ensure patient privacy while preserving the clinical content and structure. This clinical dataset includes diverse document types: anamnesis, discharge letters, laboratory reports, nursing notes, radiology reports, triage assessments, clinical diaries, home-based therapy records, medical visits, and specialist consultations. All documents are collected over a three years span at the Emergency Department of the hospital.

source	words	source	open
scientific			
Med Common Crawl	65.7M	*Med-mT5*	x
Drug Instructions	36.8M	*Med-mT5*	x
Medical Wikipedia	12.8M	*Med-mT5*	x
E3C corpus	11.6M	*Med-mT5*	x
WebHoseAZ	7.2M	*Med-mT5*	x
Thesis	5.9M	*Med-mT5*	x
Medical Websites	3.7M	*Med-mT5*	x
Pubmed	2.4M	*Med-mT5*	x
Supplement descr	1.3M	*Med-mT5*	x
others	1.0M	*Med-mT5*	x
Pensiero Scientifico	103.7M	*our*	
Unipd Theses	26.3M	*our*	x
Zadig	1.2M	*our*	
tot scientific	280.1M		
clinical			
Emergency dept.	125.7M	*our*	x
tot overall	405.8M		

Table 1: Continual Pre-Training data size by source. The *scientific* dataset is composed by several sources, resulting in 278M words. The *clinical* dataset is composed by documents coming from the emergency department of an Italian hospital, comprising 126M words. Around half of the *scientific* data is sourced from Med-mT5 (García-Ferrero et al., 2024), while the remaining is from our new data sources. All datasets are made open-source (*open*), with the exception of *Pensiero Scientifico* and *Zadig*, due to their restrictive licence.

Both datasets undergo distinct preprocessing pipelines before continual pre-training. Data from García-Ferrero et al. (2024) is already provided in a chunked textual format; we use it directly after tokenization. In contrast, the PDF-based sources (Pensiero Scientifico, University of Padova theses, and Zadig) require a multi-step pipeline. We

[4] https://thesis.unipd.it/

[5] https://thesis.unipd.it/sr/static/licenza.htm

[6] https://fenicenetwork.marionegri.it

convert each PDF into a hierarchical structured representation using the Docling library (Team, 2024), which applies document layout analysis to extract text blocks (e.g., paragraphs, list items) while preserving reading order across multi-column layouts; we disable table structure detection to reduce processing overhead. We then perform language detection on each extracted chunk using *langdetect* and retain only Italian-language chunks. We discard headers, footers, and spurious non-textual fragments (e.g., lines consisting solely of numbers or years, lines with an anomalously high punctuation-to-character ratio). We apply a normalization step common to all sources: we strip URLs via regular expressions and apply a word-spacing correction heuristic based on NLTK WordNet word-boundary lookup (Bird et al., 2009) to recover missing whitespace introduced by PDF encoding artifacts. We discard chunks shorter than 20 characters throughout. On the other hand, the *clinical dataset* is already provided in plain-text format by the hospital; we therefore bypass the PDF conversion and language-filtering stages and segment EHR records by document type before tokenization. At training time, we combine and shuffle all sources with a set seed. We split sequences exceeding the model's maximum context length into non-overlapping fixed-length chunks rather than truncating them, so as to limit information loss (Raffel et al., 2023). The replication of our experiments requires just tokenization and collation of the dataset we release, as it is distributed in its processed version.

3. Tasks and Datasets

We address five NLP tasks defined over twelve Italian datasets and twenty sub-tasks (Table 2) in the medical domain, modeled according to a unified approach. When possible, we reserve some datasets for out-of-distribution testing by not exposing models to them at training time. Rather than designing separate structures for each task, we reformulate each problem in an instruction-following format: the model receives a textual input representing the example at hand (e.g. a clinical note, a patient report, a multiple-choice question). The model's response is expected to be a JSON-formatted string, ensuring that the output is structured, consistent, and easily parsable by downstream systems.

Named Entity Recognition Named Entity Recognition (NER) involves identifying and extracting entities in text. For this task, we based our experiments on three datasets: E3C (Magnini et al., 2023), a collection of clinical narratives; PharmaER (Zugarini and Rigutini, 2025), composed of leaflets of drugs authorized by the Italian Medicines Agency; CardioCCC (Lima-López et al., 2024), a selection of cardiology clinical case reports analogous to discharge summaries. For E3C, we defined a sub-task for each annotation type between *clinical entities* and *body parts*; for PharmaER we considered *drugs*, *diseases*, *anatomical parts*, and *symptoms*; for CardioCCC we considered the *medications* annotations.
We kept the other three datasets for out-of-distribution evaluation at testing time. To assess performances on a task identical to one present in the training data (E3C), but on different clinical notes, we consider *clinical entities* in the E3C-projected version by Ghosh et al. (2025)). To test entity types seen at training time, but with slightly different definitions due to different annotation processes, we select *diseases* enitites in health records by Miranda-Escalada et al. (2022) (DisteMIST). To assess performances on entity types unseen at training type we consider *cognitive symptoms* in psychiatric records by Crema et al. (2023) (PsyNIT).

Case Report Form Filling Case Report Forms (CRFs) are structured documents used to systematically record patient information during clinical trials or routine care. The CRF filling task requires models to extract relevant information from unstructured clinical narratives and populate the corresponding fields in the forms. We used the *diagnosis*, *clinical history*, and *exams* sub-tasks from the dataset proposed by Ferrazzi et al. (2025a), which builds on E3C by mapping clinical notes to structured CRF fields. Considering this is the only available resource for the task, we could not select any dataset for out-of-distribution evaluation.

Medical Question Answering Medical Question Answering (QA) involves providing accurate answers to clinically relevant questions based on text or structured data. In this work, we focus on multiple-choice QA, where the model selects the correct answer from a set of options. We performed experiments using a dataset originating from medical exams -MedExpQA- proposed by Alonso et al. (2024). We utilized both the *plain questions* subtask and the *rag* one, where each question is enriched with relevant context.
We kept three other datasets of medical exams for out-of-distribution evaluation at testing time: Italian admission tests data by Casola et al. (2023) (AT) to test on native Italian data; a translated version of MedMCQA (Pal et al., 2022) and MedQA (Jin et al., 2021) proposed by Ferrazzi et al. (2025b) to determine performance on two of the most utilized benchmarks in the field (Pal et al., 2024).

Relation Extraction Relation Extraction (RE) consists of identifying semantic relationships between medical entities in text. For this task, we leveraged the E3C dataset, focusing on the *pertains-to* relations between exams and laboratory tests, and their results. To the best of our knowledge, this is the only available resource for the task in Italian, and we could not select any dataset for out-of-distribution evaluation.

Argument Mining Argument mining involves identifying and structuring reasoning elements or argumentative components in clinical text, such as claims, premises, markers, diseases, treatments, and diagnoses. We conducted experiments on the Casimedicos-Arg dataset (Sviridova et al., 2024), which contains annotated medical texts for argument detection. Similarly to RE and CRF filling, we could not select any dataset for out-of-distribution evaluation as Casimedicos-Arg is the only available resource for the task in the medical domain.

Evaluation metrics Named Entity Recognition, Case Report Form filling, Argument Mining, and Relation Extraction are evaluated using F1 score on exact match, while multiple-choice QA is evaluated using accuracy. For each task, we report the average among datasets and subtasks. We determine an overall score by averaging individual metrics.

Tasks definition			Examples		
task	**dataset**	**n**	**train**	**val**	**test**
ner	cardioccc	1	250	100	150
ner	pharmaer	4	1316	404	64
ner	e3c	2	451	67	628
re	e3c	1	451	67	628
arg	casimed	1	434	63	125
crf	e3c	3	2.170	404	2.615
qa	medexpqa	2	434	63	125
total	all	14	5.506	1.168	4.335
The following are used only for o-o-d testing					
ner	distemist	1	-	-	750
ner	psynit	1	-	-	400
ner	e3c-proj	1	632	101	738
qa	at	1	21	-	500
qa	medmcqa	1	182k	-	4.183
qa	medqa	1	10k	1.272	1.273
total	all	6	-	-	7.844

Table 2: Datasets selected for Italian medical NLP tasks definition: named entity recognition (*ner*); relation extraction (*re*); argument mining (*arg*); case report forms filling (*crf*); multiple-choice question answering (*qa*). The number of sub-tasks defined per dataset (*n*) is reported. The table reports datasets used at both training and testing time (*in-distribution*), together with the ones utilized solely for testing purposes (*out-of-distribution*) in the last six rows.

4. Experimental settings

We implement the strategies described in Section 2 to improve the performance of small models on medical NLP tasks in Italian. To assess how "good" they can be, we establish competitive baselines using larger state-of-the-art LLMs. This comparison allows us to quantify the performance gap between compact and large models, and to understand under which conditions small models can offer a practical, resource-efficient alternative for real-world medical applications.

Baselines We select two LLMs, Qwen3-32B and Medgemma-27B prompted via 4-shot, as baselines. Both models are chosen for their high performance on Italian general-domain tasks, following recent benchmark results (Magnini et al., 2025). We observed that the best performances are obtained by Qwen3-32B, which we therefore considered as the primary baseline (Table 3, first two rows).

Models To select the model families for evaluation, we considered several criteria: (i) whether the family includes models with approximately one billion parameters; (ii) the level of adoption within the NLP research community, as we aim to provide insights that are relevant to widely used architectures; and (iii) their performance in Italian. Based on these factors, we selected Llama-3.2-1B (Dubey et al., 2024), Gemma-3-1B (Team, 2025), and Qwen-3-1.7B (Yang et al., 2025). We utilized all models, obtaining six models overall.
Note that the way models report the number of parameters in the name differs among families: Qwen and Llama have, respectively, 72% and 24% more parameters than Gemma, which has an actual number of parameters exactly equal to one billion.

Methods to enhance performances We test and evaluate the impact of several approaches to improve performance across tasks, both at inference and training time. First, we assess models in **0-shot** and **4-shot**, using structured prompts for instructed models, and much simpler ones for base models. We then incorporate **constraint decoding**, which enforces valid output structures and reduces formatting errors during generation. We test the impact of **supervised fine-tuning**, which enables models to learn both the semantic aspects of the tasks and the expected output formats. Fi-

nally, we explore **continual pretraining**, further exposing models to large amounts of domain-specific medical data to enhance their understanding of specialized terminology and clinical language patterns.

Supervised Fine-Tuning settings To perform supervised fine-tuning on both the base and instruction-tuned models—as well as on their counterparts that underwent continual pretraining, 12 models in total—we employed the LoRA method (Hu et al., 2022). Fine-tuning was conducted with a learning rate of *5e-4* using a cosine learning rate scheduler, on a single NVIDIA H200 GPU, with a batch size of 16. On average, training required approximately two hours per model, resulting in 24 hours overall.

Continual Pre-training settings We train using the transformers library on 3 Nvidia L40S GPUs with accelerate[7]. We employed $BF16$ mixed precision training to optimize memory usage while maintaining numerical stability. In our preliminary experiments, we tested three different learning rate configurations: (1) a cosine scheduler with 30% warmup and peak learning rate of 0.0002, (2) a constant scheduler with no warmup and learning rate of 0.0002, and (3) a constant scheduler with no warmup and lower learning rate of $5e-5$ to mitigate overfitting (Ibrahim et al., 2024). We observed that the first configuration gives the best performance in terms of loss.
We utilized sequence packing techniques with Flash Attention 2 to maximize efficiency, processing sequences up to 1024 tokens with appropriate attention masking to prevent cross-sample contamination (Kundu et al., 2024a). Gradient accumulation and synchronization were handled automatically by the Accelerate library across the 3 GPUs. Each training run required approximately 8-12 hours, depending on the model size, with larger models (Qwen-3 1.7B) taking closer to 12 hours while smaller models (Gemma-3 1B) completing in around 8 hours. We exposed to CPT both base and instructed-tuned models (e.g., Llama-3.2-1B and Llama-3.2-1B-Instruct), following the findings of Sainz et al. (2025), which highlight how reverting the order of CPT first and instruction tuning second might lead to better results. All models were trained on the combined scientific and clinical datasets.
More details about CPT choices can be found in Appendix.

5. Results and Discussion

We evaluate whether small LLMs (around 1B parameters) can approximate the behaviour of larger models through targeted adaptation strategies. Table 3 summarizes the performance of Gemma-3, Qwen3, and Llama-3.2 across all configurations described in Section 2. We aggregate results by averaging the F1 scores per task (i.e., each task contributes equally to the final F1 score). In Table 10 (Appendix) we showcase that different aggregation strategies lead to the same conclusions. We do so by averaging results per sub-task, and by assigning a weight to each sub-task according to the number of testing examples. Tables 7, 8 report the results over each task.
To estimate the statistical significance of the observed results, we perform t-tests for paired observation, where each pair is composed by the performance of the baseline and of the adapted model, respectively. We calculate the p-value for the hypothesis of the delta in performance between the baseline and the adapted model being higher than zero, and report results in Table 3, last column.

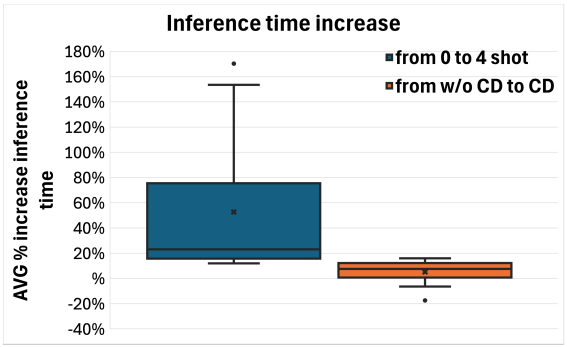


Figure 2: Impact on inference time of using *4-shot* and Constraint Decoding (*CD*) settings. While 4-shot significantly increases the time required to run the inference, CD does not. The average is calculated among 5 models and 14 subtasks, using the *vLLM* and *outlines* libraries for model serving.

Inference-time methods We first compare inference-time methods, namely constraint decoding (CD) and few-shot prompting (4-shot), and the results are shown in Figure 1 (left). Across all models, 4-shot prompting presents an average increase from 0-shot of $+9.4$ points, consistently yielding higher performance than constraint decoding ($+3.8$ average increase). This difference likely arises because few-shot examples convey not only structural but also semantic cues about the task. This gain comes at a computational cost: inference time in-

[7] https://github.com/huggingface/accelerate

Model	Method	NER	CRF	RE	QA	ARG	AVG	δ baseline	p-val
medgemma-27b	*+ 4-shot*	52.4	62.2	8.9	83.2	62.6	53.8		
Qwen3-32B	*+ 4-shot*	49.7	63.3	11.5	82.8	66.3	54.7		
gemma-3-1b-pt 1.00B params	*0-shot*	0.0	0.0	0.0	0.0	0.0	0.0	-54.7	
	+ CD	4.8	0.3	0.0	16.8	0.0	4.4	-50.3	
	+ 4-shot	3.9	13.2	3.1	23.2	1.9	9.1	-45.6	
	+ CD, 4-shot	12.6	13.3	4.1	23.2	12.4	13.1	-41.6	
	+ FT	60.1	67.1	23.4	52.4	76.5	55.9	+1.2	*
	+ CPT, FT	48.0	42.5	5.6	19.2	57.5	34.6	-20.1	
gemma-3-1b-it 1.00B params	*0-shot*	21.1	8.3	0.1	37.6	4.3	14.3	-40.4	
	+ CD	20.0	26.4	0.0	32.4	1.3	16.0	-38.7	
	+ 4-shot	16.9	25.1	0.0	32.0	17.8	18.4	-36.3	
	+ CD, 4-shot	16.9	35.8	0.6	37.2	30.5	24.2	-30.5	
	+ FT	**61.5**	**68.5**	24.0	51.6	**78.5**	56.8	+2.1	*
	+ CPT, FT	53.6	65.8	**47.2**	**61.8**	68.7	**59.4**	+4.7	*
Llama-3.2-1B 1.24B params	*0-shot*	0.0	0.0	0.0	0.0	0.0	0.0	-54.7	
	+ CD	6.8	3.3	0.0	15.6	1.2	5.4	-49.3	
	+ 4-shot	12.0	11.1	3.1	25.6	22.0	14.8	-39.9	
	+ CD, 4-shot	14.9	13.2	3.1	26.0	31.6	17.7	-37.0	
	+ FT	39.0	51.5	30.6	57.6	70.6	49.9	-4.8	
	+ CPT, FT	47.8	34.6	20.8	37.6	74.8	43.1	-11.6	
Llama-3.2-1B-Instruct 1.24B params	*0-shot*	15.8	18.1	1.3	42.8	7.2	17.0	-37.7	
	+ CD	17.2	27.9	1.3	43.2	6.6	19.2	-35.5	
	+ 4-shot	10.3	23.6	6.3	36.0	8.7	17.0	-37.7	
	+ CD, 4-shot	11.2	10.5	6.2	35.6	23.1	17.3	-37.4	
	+ FT	56.7	**71.7**	**30.9**	**59.2**	**76.6**	**59.0**	+4.3	**
	+ CPT, FT	**59.7**	64.7	27.5	43.2	73.8	53.8	-0.9	
Qwen3-1.7B-Base 1.72B params	*0-shot*	0.0	0.0	0.0	0.0	0.0	0.0	-54.7	
	+ CD	28.6	7.1	1.0	7.9	25.8	14.1	-40.6	
	+ 4-shot	30.7	29.5	2.1	12.3	50.2	25.0	-29.7	
	+ CD, 4-shot	34.8	24.3	2.6	13.3	52.7	25.5	-29.2	
	+ FT	62.2	**73.7**	33.2	60.8	79.4	61.9	+7.2	***
	+ CPT, FT	59.4	70.3	18.3	52.8	74.0	55.0	+0.3	*
Qwen3-1.7B 1.72B params	*0-shot*	18.0	31.8	4.1	56.0	11.2	24.2	-30.5	
	+ CD	18.4	35.3	4.0	56.8	11.4	25.2	-29.5	
	+ 4-shot	20.6	37.0	12.3	44.4	54.5	33.8	-20.9	
	+ CD, 4-shot	20.4	37.1	12.4	59.2	54.9	36.8	-17.9	
	+ FT	**62.6**	72.9	**37.9**	**64.0**	**82.0**	**63.9**	+9.2	***
	+ CPT, FT	61.5	67.4	25.3	57.6	77.6	57.9	+3.2	**

Table 3: Impact of different methods on the five tasks for the three selected model families. For each model family, the best and second-best performances are in **bold** and underlined respectively. The first two rows report the performances of the baseline models Qwen3-32B and medgemma-27b-text-it. The "δ baseline" column reports the delta from the best performing baseline (Qwen3-32B with 4-shot). The last column (*p-val)* reports the significance for the hypothesis of the adaptation method performances being higher than the baseline. One * refers to an observed significance higher than 0.7, ** higher than 0.8, and *** higher than 0.95. The test is performed by pairing results at a sub-task level.

creases by an average of 53% when moving from 0-shot to 4-shot configurations (statistically significant), whereas applying CD has no significant impact on runtime (Figure 2). Full results in Table 6). When the context allows, combining 4-shot prompting with constraint decoding yields the most reliable results ($+13.2$ on average).

In general, base models show greater improvements compared to their instruction-tuned counterparts ($+9.4$ and $+3.8$ respectively), which suggests that they possess untapped potential that can be effectively steered toward task-specific behaviour through adaptation. Interestingly, Llama-3.2-1B-Instruct shows no benefit, suggesting that it has already undergone an instruction-tuning phase closely aligned with the structure and objectives of our task definitions.

model	method	AVG	delta
Qwen3-32B (baseline)	4-shot	61.7	
Llama-3.2-1B-Instruct	4-shot + CD	24.1	
Llama-3.2-1B-Instruct	FT	35.9	+11.9
Qwen3-1.7B	4-shot + CD	33.2	
Qwen3-1.7B	FT	39.3	+6.1
gemma-3-1b-it	4-shot + CD	27.6	
gemma-3-1b-it	CPT + FT	31.4	+3.9

Table 4: Average performances of the adapted models and their counterparts for each of the families on Out-Of-Distribution datasets. The *delta* column represents the improvement in performance due to FT/CPT.

Training-time methods Among training-time strategies, supervised fine-tuning (FT) consistently produces the best results across all models and tasks (Figure 1, right). Fine-tuning enables models to internalize both the semantic and structural aspects of the tasks, resulting in substantial performance gains. We also evaluate continual pretraining (CPT) followed by fine-tuning. While CPT+FT achieves competitive performance and can match larger baselines, it is generally less effective than fine-tuning alone, with the notable exception of Gemma3-1b-it, where CPT provides additional benefits.

Task-level analysis A task-wise breakdown reveals consistent patterns across models. The Relation Extraction task emerges as the most challenging, indicating that smaller models still struggle with learning and reasoning over complex inter-entity relationships. In contrast, Question Answering (QA) is the easiest task, likely because models have already been exposed to large amounts of QA-style data during training. This is the only task where the baselines outperform our best adapted models. The tasks that benefits the most from adaptation is argument mining.

Best models Our experiments show that five of the six selected small LLMs can surpass large models, demonstrating that careful adaptation can compensate for model size. Our smallest model (Gemma-3-1b-it+CPT+FT) achieves an average score of 59.4, $+4.7$ above the Qwen-32B, 4-shot baseline. With 72% of the parameters more than Gemma, Qwen-3-1.7B+FT stands out, reaching an overall macro score of 63.9 and surpassing the performance of the baseline by $+9.2$ points.

6. Out-of-distribution tasks

To assess the generalization capabilities of the trained models, we evaluate the models on six datasets not utilized at training time, highlighted in Table 2. For those datasets without an official train-test split, we keep 4 examples for few shots in the baseline, and test on all the remaining ones. For each model family, we select the models with the best performances in in-distribution datasets, resulting in Llama-3.2-1B-Instruct+FT, Gemma-3-1b-it+CPT+FT, and Qwen3-1.7B+FT.

To evaluate whether training improves generalization to out-of-domain datasets, we compare each adapted model (i.e., models that underwent our FT/CPT) with its corresponding pre-adaptation version. To enforce fair comparison, for each trained model, we prompt its pre-FT/CPT version using the best-performing configuration identified earlier (4-shot with constraint decoding). We evaluate the FT/CPT models in zero-shot settings to test whether the learned knowledge transfers beyond the training distribution, and compare it to the best-performing inference-time method. The results (Table 4) show consistent improvements: Llama-3.2 achieves a gain of $+11.9$ points, Qwen3 improves by $+6.1$, and Gemma-3 by $+3.9$. Although these gains confirm that training enhances generalization, the magnitude of improvement is considerably smaller than that observed for in-distribution data. Furthermore, improvements are visible for NER sub-tasks, while neglettable for QA ones (see Table 9 for results at sub-task granularities). Consequently, when compared with the Qwen3-32B 4-shot baseline, the small models still fall short in overall performance.

In summary, while domain adaptation through fine-tuning effectively improves performance even on out-of-distribution datasets, larger models continue to generalize more robustly. These findings highlight that achieving strong generalization in specialized domains still requires task- or dataset-specific adaptation.

7. Related Work

Small Language Models Recent work has demonstrated that SLLMs can achieve competitive performance when properly adapted to specific domains (Shi et al., 2025; allal et al., 2025), highlighting their efficiency as a property that makes them particularly attractive for resource-constrained environments. García-Ferrero et al. (2024); Farzi et al. (2024) showcase how SLLMs can be effectively adapted to the medical domain.

Constraint decoding Constraint decoding consists of defining a formal grammar and enforcing

LLMs to generate accordingly (Geng et al., 2023). Park et al. (2024) have presented the potential issues and how to overcome them.

Continual Pre-training for Domain Adaptation Continual pre-training has emerged as an effective strategy for adapting large language models to specialized domains without losing general language understanding (Cossu et al., 2024; Yildiz et al., 2025). This approach is particularly valuable in medical domains, where specific terminology and reasoning patterns differ significantly from general text (Gururangan et al., 2020a). Examples of models adapted to the biomedical domain via continual pretraining are BioMedROBERTA (Gururangan et al., 2020b), ClinicalBERT (Huang et al., 2019) for encoder-only models; decoder-only models as BioGPT (Luo et al., 2022), Meditron (Chen et al., 2023); encoder-decoder models as Med-mT5 (García-Ferrero et al., 2024).

8. Conclusion

We found that overall, the best approach to adapt small LLMs to Italian medical tasks is to perform supervised fine-tuning, while continual pre-training rarely gives enhancements. If acting at inference time, combining constraint decoding with few-shot prompting offers the best results.
Overall, our experiments demonstrate that small LLMs can achieve competitive performance on a variety of medical NLP tasks when appropriately adapted, even exceeding larger baselines. Our best model, based on Qwen3-1.7B, outperforms Qwen3-32B (4-shots) by an average of $+9.2$ points. This benefit is reduced when addressing out-of-distribution data, suggesting that achieving strong performances still requires dataset-specific tuning. In conclusion, we found that small, resource-efficient LLMs emerge as a viable solution for healthcare institutions with limited resources, though more research remains crucial to fully exploit their potential in out-of-distribution medical NLP tasks.

9. Limitations

This study focuses on Italian medical NLP datasets, which may limit the generalizability of our findings to other languages. Cross-lingual transfer and multilingual adaptation were not explored and remain avenues for future research.

Our out-of-distribution evaluation was conservative: we avoided tuning the training phase on unseen data to prevent overfitting. While this ensures a fair assessment, it may underestimate model robustness across diverse domains and institutions.

We evaluated only a limited set of adaptation strategies and did not consider reinforcement learning. Additionally, the datasets exhibit task and dataset imbalances—e.g., Question Answering is relatively easier but overrepresented, whereas Relation Extraction is underrepresented and more challenging—which may bias aggregated results. Finally, out-of-distribution evaluation does not cover all task types, limiting the completeness of our generalisation analysis across the medical NLP landscape.

Acknowledgments

This work has been partially funded by the European Union under the Horizon Europe eCREAM Project (Grant Agreement No.101057726). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HADEA). Neither the European Union nor the granting authority can be held responsible for them.

Bibliographical References

Loubna Ben allal, Anton Lozhkov, Elie Bakouch, Gabriel Martin Blazquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Agustín Piqueres Lajarín, Hynek Kydlíček, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan Son NGUYEN, Ben Burtenshaw, Clémentine Fourrier, Haojun Zhao, Hugo Larcher, Mathieu Morlon, Cyril Zakka, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2025. SmolLM2: When smol goes big — data-centric training of a fully open small language model. In *Second Conference on Language Modeling*.

Iñigo Alonso, Maite Oronoz, and Rodrigo Agerri. 2024. MedExpQA: Multilingual benchmarking of large language models for medical question answering. *Artificial Intelligence in Medicine*, 155:102938.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott

Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Silvia Casola, Tiziano Labruna, Alberto Lavelli, Bernardo Magnini, et al. 2023. Testing ChatGPT for stability and reasoning: A case study using Italian medical specialty tests. In *Proceedings of the Nineth Italian Conference on Computational Linguistics*.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. Meditron-70b: Scaling medical pretraining for large language models.

Andrea Cossu, Antonio Carta, Lucia C. Passaro, Vincenzo Lomonaco, Tinne Tuytelaars, and Davide Bacciu. 2024. Continual pre-training mitigates forgetting in language and vision. *Neural Networks*, 179:106492.

Claudio Crema, Tommaso Mario Buonocore, Silvia Fostinelli, Enea Parimbelli, Federico Verde, Cira Fundarò, Marina Manera, Matteo Cotta Ramusino, Marco Capelli, Alfredo Costa, Giuliano Binetti, Riccardo Bellazzi, and Alberto Redolfi. 2023. Advancing italian biomedical information extraction with transformers-based models: Methodological insights and multicenter practical application. *Journal of Biomedical Informatics*, 148:104557.

Claudio Crema, Federico Verde, Pietro Tiraboschi, Camillo Marra, Andrea Arighi, Silvia Fostinelli, Guido Maria Giuffré, Vera Pacoova Dal Maschio, Federica L'Abbate, Federica Solca, Barbara Poletti, Vincenzo Silani, Emanuela Rotondo, Vittoria Borracci, Roberto Vimercati, Valeria Crepaldi, Emanuela Inguscio, Massimo Filippi, Francesca Caso, Alessandra Maria Rosati, Davide Quaranta, Giuliano Binetti, Ilaria Pagnoni, Manuela Morreale, Francesca Burgio, Michelangelo Stanzani-Maserati, Sabina Capellari, Matteo Pardini, Nicola Girtler, Federica Piras, Fabrizio Piras, Stefania Lalli, Elena Perdixi, Gemma Lombardi, Sonia Di Tella, Alfredo Costa, Marco Capelli, Cira Fundarò, Marina Manera, Cristina Muscio, Elisa Pellencin, Raffaele Lodi, Fabrizio Tagliavini, and Alberto Redolfi. 2024. Medical information extraction with NLP-powered QABots: A real-world scenario. *IEEE Journal of Biomedical and Health Informatics*, 28(11):6906–6917.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The Llama 3 herd of models. *CoRR*, abs/2407.21783.

Saeed Farzi, Soumitra Ghosh, Alberto Lavelli, and Bernardo Magnini. 2024. Get the best out of 1B LLMs: Insights from information extraction

on clinical documents. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, Bangkok, Thailand. Association for Computational Linguistics.

Pietro Ferrazzi, Alberto Lavelli, and Bernardo Magnini. 2025a. Converting annotated clinical cases into structured case report forms. In *Proceedings of the 24th Workshop on Biomedical Language Processing*, Vienna, Austria. Association for Computational Linguistics.

Pietro Ferrazzi, Aitor Soroa, and Rodrigo Agerri. 2025b. Grounded multilingual medical reasoning for question answering with large language models. *CoRR*, abs/2512.05658.

Iker García-Ferrero, Rodrigo Agerri, Aitziber Atutxa Salazar, Elena Cabrio, Iker de la Iglesia, Alberto Lavelli, Bernardo Magnini, Benjamin Molinet, Johana Ramirez-Romero, German Rigau, Jose Maria Villa-Gonzalez, Serena Villata, and Andrea Zaninello. 2024. MedMT5: An open-source multilingual text-to-text LLM for the medical domain. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Torino, Italia. ELRA and ICCL.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured NLP tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10932–10952. Association for Computational Linguistics.

Soumitra Ghosh, Begoña Altuna, Saeed Farzi, Pietro Ferrazzi, Alberto Lavelli, Giulia Mezzanotte, Manuela Speranza, and Bernardo Magnini. 2025. Low-resource information extraction with the European Clinical Case Corpus. *CoRR*, abs/2503.20568.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020a. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8342–8360. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020b. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. ClinicalBERT: Modeling clinical notes and predicting hospital readmission. *CoRR*, abs/1904.05342.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L. Richter, Quentin Gregory Anthony, Eugene Belilovsky, Timothée Lesort, and Irina Rish. 2024. Simple and scalable strategies to continually pre-train large language models. *Trans. Mach. Learn. Res.*, 2024.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11:6421.

Achintya Kundu, Rhui Dih Lee, Laura Wynter, Raghu Kiran Ganti, and Mayank Mishra. 2024a. Enhancing training efficiency using packing with flash attention. *CoRR*, abs/2407.09105.

Achintya Kundu, Rhui Dih Lee, Laura Wynter, Raghu Kiran Ganti, and Mayank Mishra. 2024b. Enhancing training efficiency using packing with flash attention.

Salvador Lima-López, Eulàlia Farré-Maduell, Jan Rodríguez-Miret, Miguel Rodríguez-Ortega, Livia Lilli, Jacopo Lenkowicz, Giovanna Ceroni, Jonathan Kossoff, Anoop Shah, Anastasios Nentidis, Anastasia Krithara, Georgios Katsimpras, Georgios Paliouras, and Martin Krallinger. 2024. Overview of MultiCardioNER task at BioASQ 2024 on medical specialty and language adaptation of clinical ner systems for Spanish, English and Italian. In *Conference and Labs of the Evaluation Forum*.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6).

Bernardo Magnini, Begoña Altuna, Alberto Lavelli, Anne-Lyse Minard, Manuela Speranza, and Roberto Zanoli. 2023. *European Clinical Case Corpus*, pages 283–288. Springer International Publishing, Cham.

Bernardo Magnini, Roberto Zanoli, Michele Resta, Martin Cimmino, Paolo Albano, Marco Madeddu,

and Viviana Patti. 2025. Evalita-LLM: Benchmarking large language models on Italian. *CoRR*, abs/2502.02289.

Antonio Miranda-Escalada, Luis Gasco, Salvador Lima-López, Eulàlia Farré-Maduell, Darryl Estrada, Anastasios Nentidis, Anastasia Krithara, Georgios Katsimpras, Georgios Paliouras, and Martin Krallinger. 2022. Overview of DisTEMIST at BioASQ: Automatic detection and normalization of diseases from clinical texts: results, methods, evaluation and multilingual resources. In *Working Notes of Conference and Labs of the Evaluation (CLEF) Forum. CEUR Workshop Proceedings*.

Ankit Pal, Pasquale Minervini, Andreas Geert Motzfeldt, Aryo Pradipta Gema, and Beatrice Alex. 2024. openlifescienceai/open-medical-llm-leaderboard. https://huggingface.co/spaces/openlifescienceai/open-medical-llm-leaderboard.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Kanghee Park, Jiayu Wang, Taylor Berg-Kirkpatrick, Nadia Polikarpova, and Loris D' Antoni. 2024. Grammar-aligned decoding. In *Advances in Neural Information Processing Systems*, volume 37, pages 24547–24568. Curran Associates, Inc.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Oscar Sainz, Naiara Perez, Julen Etxaniz, Joseba Fernandez de Landa, Itziar Aldabe, Iker García-Ferrero, Aimar Zabala, Ekhi Azurmendi, German Rigau, Eneko Agirre, Mikel Artetxe, and Aitor Soroa. 2025. Instructing large language models for low-resource languages: A systematic study for Basque. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29136–29160, Suzhou, China. Association for Computational Linguistics.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. 2025. Continual learning of large language models: A comprehensive survey. *ACM Comput. Surv.*, 58(5).

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. 2026. Continual learning of large language models: A comprehensive survey. *ACM Comput. Surv.*, 58(5).

Ekaterina Sviridova, Anar Yeginbergen, Ainara Estarrona, Elena Cabrio, Serena Villata, and Rodrigo Agerri. 2024. CasiMedicos-Arg: A Medical Question Answering Dataset Annotated with Explanatory Argumentative Structures. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18463–18475.

Deep Search Team. 2024. Docling technical report. Technical report.

Gemma Team. 2025. Gemma 3 technical report. *CoRR*, abs/2503.19786.

Fali Wang, Minhua Lin, Yao Ma, Hui Liu, Qi He, Xianfeng Tang, Jiliang Tang, Jian Pei, and Suhang Wang. 2025. A survey on small language models in the era of large language models: Architecture, capabilities, and trustworthiness. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 6173–6183, New York, NY, USA. Association for Computing Machinery.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.

Çagatay Yildiz, Nishaanth Kanna Ravichandran, Nitin Sharma, Matthias Bethge, and Beyza Ermis. 2025. Investigating continual pretraining in large language models: Insights and implications. *Trans. Mach. Learn. Res.*, 2025.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Guoyin Wang, and Fei Wu. 2026. Instruction tuning for large language models: A survey. *ACM Comput. Surv.*, 58(7).

Andrea Zugarini and Leonardo Rigutini. 2025. PharmaER.IT: an Italian dataset for entity recognition in the pharmaceutical domain. In *Proceedings of the Eleventh Italian Conference on Computational Linguistics*.

Appendix: Pretraining Details

This appendix describes the integration of Flash Attention 2 with sequence packing to improve the efficiency of continual pre-training. We outline the main characteristics of Flash Attention 2, explain how sequence packing reduces computational waste compared to traditional padding, and describe the attention masking mechanism that ensures proper isolation between packed sequences.

Flash Attention 2: Technical Overview

Flash Attention 2 builds upon the original Flash Attention algorithm (Dao et al., 2022), introducing several optimizations that reduce memory traffic between GPU hierarchies while preserving the exact results of standard attention (Dao, 2024).

The algorithm relies on three main techniques: **(i) Tiling**, which partitions computations into blocks that fit in the GPU's on-chip memory (SRAM), minimizing global memory access; **(ii) Recomputation**, which discards intermediate results during the forward pass and recomputes them in the backward pass to save memory; and **(iii) Kernel fusion**, which merges multiple GPU operations into single kernels to reduce launch overhead.

These optimizations result in faster trainings and lower memory consumption compared to standard implementations, while maintaining numerical stability and convergence behavior.

Sequence packing vs. Traditional padding

In conventional training pipelines, sequences within the same mini-batch are padded to a uniform length, forcing the model to process large numbers of padding tokens, which carry no information. Sequence packing mitigates this inefficiency by concatenating multiple variable-length sequences into a single tensor while storing the information about their boundaries.

This method eliminates the computational cost of padding tokens, increases the ratio of meaningful tokens per batch, and improves GPU memory utilization. It was particularly helpful for our Italian medical datasets, where even small efficiency gains have a large cumulative impact, since many trainings were performed to find the best configuration of hyperparameters. This enables efficient processing of sequences up to 1024 tokens on three Nvidia L40S GPUs, maintaining sample independence throughout training.

Data collators and Attention masking

We used **PaddingFreeCollator** (Kundu et al., 2024b), which generalizes the concept of sequence packing through a dynamic batching strategy.

Flash Attention 2 supports 2D attention masks that define explicit boundaries between packed sequences. These masks define precise attention boundaries, ensuring that tokens can only attend to others within the same original sequence. This prevents any interaction between different samples in the packed tensor, while maintaining proper causal masking for autoregressive training.

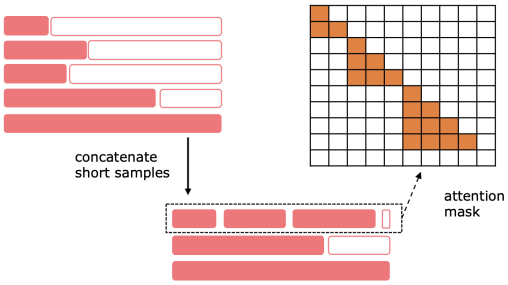


Figure 3: Visualization of sequence packing and the corresponding 2D attention mask that prevents cross-sample attention.

Implementation and Results

Flash Attention 2 was enabled in the *transformers* library by setting `attn_implementation` to `flash_attention_2` on model instantiation. Combined with the PaddingFreeCollator, this setup allowed for efficient concatenation of sequences while enforcing correct attention masking.

The resulting configuration allows continual pretraining runs to require 8 to 12 hours depending on model size, demonstrating the scalability and efficiency of this approach for domain adaptation tasks even on our reduced setup.

	4-shot vs 0-shot		CD vs w/o CD		AVG
	w/o CD	CD	0-shot	4-shot	all
gemma-3-1b-pt	9.1	8.7	4.4	4.0	6.6
gemma-3-1b-it	4.1	8.2	1.7	**5.8**	5.0
Llama-3.2-1B	14.8	**12.3**	5.4	2.9	8.9
Llama-3.2-1B-Instruct	0.0	-1.9	2.2	0.3	0.2
Qwen3-1.7B-Base	**25.0**	11.4	**14.1**	0.5	12.8
Qwen3-1.7B	9.6	11.6	1.0	3.0	6.3
AVG	10.4	8.4	4.8	2.8	

Table 5: Analysis of the impact of different methods to enhance performances at inference time. The first two columns (*4-shot vs 0-shot*) report the increase of performances moving from 0- to 4-shot with and without Constraint Decoding, while the other two (*CD vs w/o CD*) the increase of performances moving from not using Constraint Decoding to using it in both 0- and 4- shot.

Model	0-shot	0-shot + CD	4-shot	4-shot + CD
gemma-3-1b-pt	385	360	488	403
gemma-3-1b-it	194	202	227	234
Llama-3.2-1B	113	131	135	151
Llama-3.2-1B-Instruct	91	103	136	152
Qwen3-1.7B	64	71	173	180

Table 6: Inference time (in seconds) across different configurations for five selected models. Reported time refers to the execution of inference over the 14 sub-tasks on a single H100, serving them through the *vLLM* library with *outlines* for constraint decoding. Overall, inference results in around one hour. A notable difference can be observed among models, some being much faster than others. This is primarily due to the different average output length, as well as the differences in structures.

Model	Method	Named Entity Recognition						
		cardioccc	**e3c**			**pharmaer**		
		medication	*bodypart*	*clinical*	*anat. part*	*disease*	*drugs*	*symptoms*
gemma-3-1b base	0-shot	0	0	0	0	0	0	0
	+ CD	12.98	0.95	9.99	0	0	9.92	0
	+ 4-shot	5.78	0	0	0	2.67	15.65	3.51
	+ CD, 4-shot	26.01	0	4.42	0	10.2	32.86	14.93
	+ FT	99.9	32.64	57.55	67.35	48	63.93	51.55
	+ CPT, FT	70.46	27.72	48.99	52.38	35.4	60	41.38
gemma-3-1b instruct	0-shot	32.5	2.49	20.08	14.14	13.66	41.96	23.16
	+ CD	27.25	2.55	18.38	15.83	16.38	38.26	21.67
	+ 4-shot	41.68	4.63	15.63	3.39	10.75	29.49	12.7
	+ CD, 4-shot	41.96	4.51	15.85	3.39	10.64	32.05	9.68
	+ FT	99.95	33.22	58.59	72	48.85	65.14	52.94
	+ CPT, FT	99.22	31.86	24.08	54.29	56.68	50.49	58.37
Llama-3.2-1B base	0-shot	0	0	0	0	0	0	0
	+ CD	11.13	0	0.43	0	0	17.91	17.95
	+ 4-shot	32.52	0	6.9	0	9.52	31.58	3.28
	+ CD, 4-shot	33.96	0.86	9.61	0	8.89	32.84	17.91
	+ FT	19.29	9.28	28.14	62.5	46.43	61.54	46.15
	+ CPT, FT	97.79	7.61	34.74	49.35	48.98	58.93	37.21
Llama-3.2-1B instruct	0-shot	33.39	1.43	14.01	11.94	3.41	29.15	17.19
	+ CD	33.68	1.55	13.07	9	8.96	36.67	17.48
	+ 4-shot	29.96	3.72	14.42	0	7.79	16.3	0
	+ CD, 4-shot	29.05	3.93	15.11	3.7	10.26	16.42	0
	+ FT	97.95	22.14	56.01	64.71	52.94	56.76	46.43
	+ CPT, FT	99.95	28.31	58.45	66	54.26	59.23	51.92
Qwen3-1.7B base	0-shot	0	0	0	0	0	0	0
	+ CD	51	0	0	39.2	72.8	21.57	15.79
	+ 4-shot	54	26.29	17.45	35.2	59.2	6.45	16.22
	+ CD, 4-shot	54.2	26.83	26.11	37.6	75.2	8.82	14.95
	+ FT	99.95	38.19	59.07	72.07	55.88	61.95	48.15
	+ CPT, FT	99.52	23.89	57.18	72.38	60	53.39	49.48
Qwen3-1.7B instruct	0-shot	55.51	1.49	29.02	3.85	0	28.99	6.9
	+ CD	55.12	1.49	28.96	3.85	0	32.65	6.9
	+ 4-shot	59.73	3.33	29.01	13.33	3.03	35.76	0
	+ CD, 4-shot	59.25	3.33	29.07	13.33	3.03	34.62	0
	+ FT	100	34	59.78	81.9	57.55	55.42	49.6
	+ CPT, FT	100	38.13	60.26	69.9	57.55	56.54	48

Table 7: Results (F1 score) per subtask in Named Entity Recognition (NER)

		Question Answering, Relation Extraction, CRF, Argument Mining						
Model	**Method**	**QA** (medexpqa)		**RE** (e3c)	**CRF** (e3c)			**ARG**
		basic	*expert ctx*	*pertainsto*	*diagnosis*	*history*	*rml*	*arglong*
gemma-3-1b base	0-shot	0	0	0	0	0	0	0
	+ CD	16	17.6	0	0	0.44	0.56	0
	+ 4-shot	25.6	20.8	3.12	14.09	18.21	7.43	1.91
	+CD, 4-shot	25.6	20.8	4.11	14.17	18.26	7.43	12.41
	+ FT	39.2	65.6	23.42	80.95	66.67	53.59	76.5
	+ CPT, FT	16.8	21.6	5.56	65.82	44.64	17.09	57.51
gemma-3-1b instruct	0-shot	28	47.2	0.13	9.24	13.57	2	4.25
	+ CD	20	44.8	0	67.53	11.02	0.63	1.25
	+ 4-shot	27.2	36.8	0	31.96	25.56	17.73	17.8
	+CD, 4-shot	28.8	45.6	0.57	65.57	24.03	17.92	30.5
	+ FT	37.6	65.6	24	81.93	67.43	56.25	78.5
	+ CPT, FT	52	71.7	47.2	82.76	70.83	43.75	68.75
Llama-3.2-1B base	0-shot	0	0	0	0	0	0	0
	+ CD	12.8	18.4	0	9.47	0.53	0	1.15
	+ 4-shot	24	27.2	3.15	14.25	15.75	3.39	22.04
	+CD, 4-shot	24.8	27.2	3.08	20.07	13.61	5.88	31.58
	+ FT	47.2	68	30.56	41.03	65.57	47.87	70.56
	+ CPT, FT	40	35.2	20.81	10.37	78.01	15.3	74.78
Llama-3.2-1B instruct	0-shot	31.2	54.4	1.31	4.66	48.24	1.33	7.19
	+ CD	32.8	53.6	1.25	35.71	46.59	1.34	6.63
	+ 4-shot	31.2	40.8	6.27	39.25	26.97	4.55	8.68
	+CD, 4-shot	30.4	40.8	6.24	0	26.97	4.57	23.09
	+ FT	49.6	68.8	30.88	85.71	68.39	60.98	76.58
	+ CPT, FT	37.6	48.8	27.45	80.9	65.91	47.41	73.79
Qwen3-1.7B base	0-shot	0	0	0	0	0	0	0
	+ CD	15.8	0	1.04	8.03	13.3	0	25.75
	+ 4-shot	16.99	7.54	2.12	56.06	15.57	17.01	50.24
	+CD, 4-shot	18.69	7.88	2.61	56.84	15.95	0	52.74
	+ FT	41.6	80	33.17	87.06	73.49	60.61	79.39
	+ CPT, FT	38.4	67.2	18.28	80.49	69.84	60.61	73.98
Qwen3-1.7B instruct	0-shot	36	76	4.09	60.34	35.1	0	11.16
	+ CD	38.4	75.2	3.98	70.71	35.1	0	11.36
	+ 4-shot	33.6	55.2	12.27	60.74	39.71	10.42	54.52
	+CD, 4-shot	39.2	79.2	12.4	61.19	39.61	10.42	54.91
	+ FT	47.2	80.8	37.86	86.67	69.32	62.58	81.97
	+ CPT, FT	44.8	70.4	25.33	82.93	71.43	47.74	77.61

Table 8: Results (F1 score for RE, CRF, ARG; accuracy for QA) per subtask for Queston Answering (QA), Relation Extraction (RE), Case Report Form filling (CRF) and Argument Mining (ARG).

		Out Of Distribution					
Model	**Method**	**QA**			**NER**		
		at 2023	*medmcqa*	*medqa*	*e3c clinical*	*distemist*	*psynit*
gemma-3-1b instruct	0-shot	28.0	27.5	28.1	25.5	21.5	17.9
	+ CD	26.8	30.2	29.9	22.9	21.1	22.0
	+ 4-shot	27.0	27.2	30.8	27.0	23.1	25.1
	+CD, 4-shot	28.4	28.5	31.5	27.4	25.7	23.8
	+ CPT, FT	28.4	27.1	31.8	54.5	37.5	27.2
Llama-3.2-1B instruct	0-shot	29.0	29.7	31.6	17.3	25.7	12.1
	+ CD	27.6	29.4	31.5	16.7	25.4	11.9
	+ 4-shot	28.6	29.1	30.1	23.4	16.2	13.7
	+CD, 4-shot	29.0	29.1	30.6	25.3	16.6	13.0
	+ FT	31.0	27.4	30.2	51.4	43.6	31.9
Qwen3-1.7B base	0-shot	0.2	0.4	0.5	2.3	22.8	25.8
	+ CD	48.6	36.6	35.9	32.1	19.1	25.8
	+ 4-shot	39.0	29.2	35.4	31.0	24.8	27.1
	+CD, 4-shot	51.8	37.7	36.1	32.8	15.7	25.3
	+ FT	39.4	28.6	33.9	53.8	44.7	35.2

Table 9: Results (F1 score for NER; accuracy for QA) per subtask for Question Answering (QA) and Named Entity Recognition (NER).

Model	Method	AVG F1 (δ baseline) *sample-weight*	*per sub-task*	*per task*
Qwen3-32B	+ *4-shot*	52.0	56.8	54.7
gemma-3-1b base 1.00B params	*0-shot*	0.0 -52.0	0.0 -56.8	0.0 -54.7
	+ *CD*	2.8 -49.2	4.9 -51.9	4.4 -50.4
	+ *4-shot*	8.4 -43.5	8.5 -48.3	9.1 -45.7
	+ *CD, 4-shot*	10.4 -41.6	13.7 -43.2	13.1 -41.6
	+ *FT*	56.5 +4.6	59.1 +2.3	55.9 +1.2
	+ *CPT, FT*	37.3 -14.6	40.4 -16.4	34.6 -20.2
gemma-3-1b instruct 1.00B params	*0-shot*	10.8 -41.2	18.0-38.8	14.3 -40.5
	+ *CD*	18.9 -33.0	20.4 -36.4	16.0 -38.7
	+ *4-shot*	18.6 -33.4	19.7 -37.1	18.4 -36.4
	+ *CD, 4-shot*	24.5-27.5	23.6 -33.2	24.2 -30.5
	+ *FT*	**57.6** +5.7	**60.1** +3.3	56.8 +2.1
	+ *CPT, FT*	54.7 +2.8	58.0 +1.2	**59.4** +4.7
Llama-3.2-1B base 1.24B params	*0-shot*	0.0 -56.8	0.0 -56.8	0.0 -54.7
	+ *CD*	3.2 -50.4	6.4 -50.4	5.4 -49.4
	+ *4-shot*	9.9 -43.0	13.8 -43.0	14.8 -40.0
	+ *CD, 4-shot*	11.8 -40.4	15.0 -40.4	17.7 -37.0
	+ *FT*	41.0 -10.9	46.0 -10.8	49.9 -4.9
	+ *CPT, FT*	33.3 -18.7	43.5 -13.3	43.1 -11.6
Llama-3.2-1B instruct 1.24B params	*0-shot*	14.8 -37.2	18.5 -38.3	17.0 -37.7
	+ *CD*	19.6 -32.3	21.3 -35.5	19.2 -35.5
	+ *4-shot*	14.8 -34.4	18.5 -40.4	17.0 -37.8
	+ *CD, 4-shot*	11.7 -40.3	15.0 -41.8	17.3 -37.4
	+ *FT*	**58.4** +6.4	**59.8** +3.0	**59.0** +4.3
	+ *CPT, FT*	55.0 +3.0	57.1 +0.3	53.8 -1.0
Qwen3-1.7B base 1.72B params	*0-shot*	0.0 -52.0	0.0 -56.8	0.0 -54.7
	+ *CD*	7.9 -44.1	18.9 -37.9	14.1 -40.7
	+ *4-shot*	24.8 -27.2	27.2 -29.6	25.0 -29.8
	+ *CD, 4-shot*	23.7 -28.3	28.5 -28.4	25.5 -29.2
	+ *FT*	62.4 +10.4	63.6 +6.8	61.9 +7.1
	+ *CPT, FT*	56.4 +4.4	58.9 +2.1	55.0 +0.2
Qwen3-1.7B instruct 1.72B params	*0-shot*	24.8 -27.2	24.9 -31.9	24.2 -30.5
	+ *CD*	26.5 -25.5	26.0 -30.8	25.2 -29.6
	+ *4-shot*	29.2 -22.7	29.3 -27.5	33.8 -21.0
	+ *CD, 4-shot*	20.0-22.0	31.4 -25.4	36.8 -17.9
	+ *FT*	**62.4** +10.4	**64.6** +7.8	**63.9** +9.1
	+ *CPT, FT*	58.2 +6.2	60.8 +3.9	57.9 +3.1

Table 10: Average performances of models and adaptation techniques by aggregation strategy. The five tasks are composed by 14 subtasks, unevenly distridubets. Each subtask is the combination of a task and a dataset. Each dataset has a different number of testing examples. We report the F1 score averaged by number of examples in each task (*sample-weight*), by number of sub-tasks per task, and by tasks. For each model family, the best and second-best performances are in **bold** and underlined respectively. It can be noticed that there is not significant difference among aggregation method.