Adapting a Large Language Model to the Legal Domain: A Case Study in Italian

Flavio Valerio[1,†], Pierpaolo Basile[1,2,*,†] and Marco de Gemmis[1,2,*,†]

[1]Department of Computer Science, University of Bari Aldo Moro, Bari (Italy)
[2]AI2B srl, Spin-Off of the University of Bari Aldo Moro, Via E. Orabona, 4 - Bari (Italy)

Abstract

This work presents a methodology for adapting an open Large Language Model (LLM) to the Italian legal domain. We construct a legal document corpus from the Normattiva website and develop a custom scraper to ensure high-quality text extraction. The resulting corpus is used to adapt the `Llama-3.1-8b` model through continuous pre-training and Low-Rank Adaptation (LoRA). The adapted model's performance is evaluated by assessing its ability to complete sentences coherently within the new domain. Results demonstrate that the adapted model surpasses the original model across all metrics, considering various prompt lengths and different sizes of the training corpus.

Keywords

Large Language Models, Legal, Artificial Intelligence, Public Administration

1. Background and Motivation

Large Language Models have proven effective in understanding and generating text in several domains. However, language in some domains is characterized by specific structure or word usage. For example, the legal domain relies on precise language, nuanced interpretation of laws, and a vast body of evolving jurisprudence. Typical LLMs are trained on an extensive collection of documents from several domains, which can affect the capability of understanding and generating text in a specific context, such as the legal domain. Moreover, some languages are less represented than others, and the legal domain of a particular language probably needs to be added. This is a critical issue since the legal language is strongly dependent on the legislation of the specific country. Finally, using a closed LLM can be critical in a public administration domain, and then an adaptation of an open LLM can be the only alternative.

Recent works have investigated the usage of LLMs in legal domains. In [2], authors propose a few-shot entity relation extraction method in the legal domain based on large language models without training the model on domain-dependent data. In [3], several LLMs are tested on a specific dataset related to numerical estimation in the legal domain. Similarly, [4] evaluate chatGPT performance in semantic annotation of legal texts, finding that also, in zero-shot, the model can provide promising results. Following the same idea, authors in [5] evaluate the performance of chatGPT in the context of legal argument mining and underline the importance of formulating the correct prompt and how it impacts the overall performance. However, all previous works investigate close LLMs and do not consider fine-tuning or adapting existing open LLMs to the legal domain. Less recent works considered the training of the BERT-like language models specific to the legal domain or fine-tuning BERT-like models to specific legal takes [6]. Also for the Italian, a BERT model trained on Italian documents has been proposed [7]. However, these works are outside the scope of this paper since we want to focus on large language models.

To overcome these limitations, this work proposes adapting an open LLM (`Meta Llama-3.1`) to the Italian legal domain. To pursue this goal, we have created a corpus of documents by collecting legal

NL4AI 2024: Eighth Workshop on Natural Language for Artificial Intelligence, November 26-27th, 2024, Bolzano, Italy [1]
*Corresponding author.
†These authors contributed equally.
f.valerio6@studenti.uniba.it (F. Valerio); pierpaolo.basile@uniba.it (P. Basile); marco.degemmis@uniba.it (M. d. Gemmis)
0000-0002-0545-1105 (P. Basile); 0000-0002-2007-9559 (M. d. Gemmis)

texts written in Italian. Then, using an adaptation strategy, we continue training the LLM on the corpus of collected documents. To evaluate the effectiveness of the proposed approach, we measure the quality and coherence of the generated text before and after the training.

The paper is structured as follows: Section 2 describes the construction process of the legal corpus, while Section 3 provides methodological details about the adaptation of the LLM to the legal domain. Section 4 describes the evaluation and discusses the results and Section 5 closes the paper with final remarks and proposes future research directions.

2. Corpus creation

A suitable corpus of documents is necessary to improve the capabilities of an existing LLM in understanding and producing the language used in the Italian legal domain. However, we create a new one due to the absence of a publicly available dataset. To this end, we develop a web crawler for the Normattiva[1] website. This Italian website is an essential resource for the consultation of current legislation, offering access to national laws, decrees, and regulations. In addition, advanced search capabilities and multivigency consultation of acts are provided.

A crawler is a software designed to browse web pages and gather information systematically. Its applications range from search engine indexing to data analysis to content updating. Designing an effective crawler requires a customized approach tailored to the site of interest. In our case, crawler development began with an in-depth study of the structure of the Normattiva site to understand how to structure the data collection process.

The Normattiva site has two pages relevant to our purpose: pages that serve as "containers of useful links" and pages containing legislative texts of interest. The crawling requires three steps:

1. **Collection of links**: The first step is to identify a "main page" containing relevant links, such as articles of the Italian Constitution or legal acts. Once this page is identified, links are collected. To optimize the process, it is necessary to examine the page through inspection tools to identify sections (e.g., `divs`) containing links of interest. This approach avoided irrelevant sections of the page, thus reducing the time required for scraping and post-processing the data. Without such preliminary analysis, a crawler would have had to take a more "raw" approach, analyzing the entire page and producing a less refined output, which would have required further post-processing.
2. **Text Capture**: Pages containing legislative texts on Normattiva feature a sidebar on the left side that allows users to navigate between articles or legal acts via calls to a JavaScript function. This function dynamically modifies the displayed content. This feature affects the choice of libraries and approaches taken in designing the crawler. Once again, the inspection tool is used to locate the specific `div` containing the relevant text. We do not download the entire page, as this would have captured a great deal of irrelevant information, requiring post-processing, a step that was preferred to avoid.
3. **Saving information**: The information extracted by the crawler is saved in a JSON lines format. Each line in the file contains a JSON object with the following fields: `text`, `url`, `timestamp` and `source`.

The main libraries used to implement the crawler are Selenium[2] and Beautiful Soup[3]. **Selenium** is an advanced browser automation tool widely used in academia and industry to perform automated testing and manage complex web operations. Because of its versatility, Selenium supports a wide range of platforms, browsers, and programming languages, enabling the precise simulation of a real user's actions, such as selecting links, entering text, and interacting with dynamic elements through mouse clicks. An essential feature of Selenium is its ability to interact with JavaScript, a crucial aspect of managing dynamic Web sites. In developing the Normattiva crawler, Selenium is crucial for automating

[1] https://www.normattiva.it/
[2] https://www.selenium.dev/
[3] https://pypi.org/project/beautifulsoup4/

navigation and interaction with dynamic content, thus ensuring accurate and efficient capture of legislative data. For example, it allows clicking on links to move from one article to another, waiting for page elements to load properly before performing further actions. **Beautiful soap** is a library for parsing and extracting data from HTML and XML documents. It is praised for its ease of use and the intuitive interface for navigating and manipulating the structure of HTML documents. Beautiful Soup makes it possible to identify and extract structured data from web pages accurately. In our work, Beautiful Soup is used together with Selenium: Selenium handles the loading and interaction with dynamic page elements, while Beautiful Soup parses HTML source code to extract relevant textual content.

The Normattiva crawler[4] enables the systematic and targeted collection of legislative documents from the website, providing a suitable corpus for fine-tuning and adapting an existing LLM to the legal domain. The tailoring approach adopted, based on a preliminary analysis of web page structure, ensured greater efficiency than more generic methods, reducing the workload required for cleaning and processing the collected data. The final corpus[5] contains 396,592 text passages extracted from the Normattiva website for a total of about 108 million occurrences[6].

3. Large Language Model Adaptation

LLMs have demonstrated remarkable capabilities across various natural language processing tasks and languages. However, their performance is often limited when applied to specialized domains with distinct terminology, unique stylistic features, or specific contextual knowledge not appropriately covered in the training data. To bridge this gap, it is essential to adapt an LLM to new domains by exposing them to domain-specific data. A prominent approach to this adaptation is continuous pre-training using domain-specific corpora, enhanced by parameter-efficient techniques such as Low-Rank Adaptation (LoRA) [8]. We have already successfully investigated this approach in adapting BLOOM [9] and LLaMA-2 [10] and LLaMA-3 [11] models to the Italian language [12, 13, 14].

LoRA is designed to fine-tune pre-trained LLMs with minimal additional computational and memory overhead. The main idea behind LoRA is to introduce low-rank matrices into the architecture of the LLM during fine-tuning. These matrices capture task-specific or domain-specific adaptations while keeping most original model parameters frozen. This method is particularly advantageous when computational resources are limited or when there is a need to preserve the general knowledge embedded in the original LLM while introducing domain-specific knowledge. These characteristics are critical in our approach since we can introduce new knowledge related to the new domain (legal) with a low computational cost.

Our methodology is based on continuous pre-training. This process involves sequentially fine-tuning the LLM on a corpus of documents relevant to the target domain. The corpus is carefully curated to reflect the domain's linguistic patterns, terminologies, and contextual nuances. This process can be iterative, allowing the model to gradually adapt to the new domain while retaining its ability to perform well on general tasks. Moreover, we can update the model with new knowledge when, for example, new laws are added or removed. Moreover, LoRA is a Parameter-Efficient Fine-Tuning (PEFT) [15] technique and then works on a subset of the original parameters during the training. This allows the release of only the weights modified during the training, reducing the space needed to store the model. This approach makes it possible to adapt the original model on several domains by performing different LoRA training steps and producing an adapter for each. The adapter stores only the weights modified in each training step. Each adapter is interchangeable and can be loaded upon the original model.

In this work, we start from the LLaMA-3 8 billion model. We selected this model to adapt a state-of-the-art model using reasonable computing resources. In detail, all the training process is performed on

[4]The crawler is available on GitHub: https://github.com/FValerio96/NormattivaCrawling/

[5]The corpus is available on Hugging Face: https://huggingface.co/datasets/swap-uniba/normattiva-dump.

[6]Words are counted considering sequences of alphanumeric characters. The exact number of tokens depends on the specific LLM tokenizer.

a single GPU[7]. To reduce the computation cost we use the unsloth[8] library. We fine-tune the model using LoRA with $rank = 16$ and $alpha = 32$, considering all the linear layers with a max sequence of 2,048 tokens. The corpus built according to Section 2 is used to feed the training with text samples with a batch size of 16 and an accumulation step of 2. The model is trained for one epoch due to the large number of examples. The output of the training process is an adapter that can be loaded upon the `LLaMA-3.1` model. Finally, we evaluate the quality of the training process according to the experimental setting described in Section 4.

4. Evaluation

In this section, a comparative evaluation is conducted between the `Llama-3.1` model and its fine-tuned version `Llama-3.1-NA`. The experiment examines the models' generative capabilities using a partial prompt approach. Sentences from the test dataset are partially used as input (prompts) for the models, which subsequently generated the full text from this initial portion. The results are stored in JSON files containing three distinct fields:

1. **text**: the original full sentence;
2. **prompt**: the initial portion of the text provided to the model;
3. **generated**: the complete output generated by the model from the prompt.

We build two fine-tuned models by training the model on two different portions of the corpus. The model `Llama-3.1-NA-100k`[9] is trained on 100,000 text passages randomly selected from the corpus, while the model `Llama-3.1-NA`[10]exploits the whole corpus.

We randomly select 1,000 other sentences for the evaluation. For the evaluation, each testing sentence is tokenized using the model tokenizer, and we retain only the first k tokens for each sentence as the prompt. In our case, the sentence is removed if it exceeds the maximum input length, 2,048 tokens.

Finally, all models were used to complete the generated texts. We refer to the base model using the label `Llama3.1`.

For the evaluation, the generated text is compared against the original text by using three metrics:

1. **BLEU** (Bilingual Evaluation Understudy) is a metric that quantifies the similarity between the text generated by a model and a reference text, using the geometric mean of the n-grams shared between the two texts.
2. **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics that assess the quality of generated texts, particularly summaries, by comparing them with reference texts. Common variants include ROUGE-N, which measures the correspondence of n-grams, ROUGE-L, which considers the longest common sub-sequences, and ROUGE-W, which takes into account the weight of correspondences.
3. **BERTScore** relies on pre-trained language models to assess the semantic similarity between the generated and reference texts, going beyond mere superficial word matching.
4. **Perplexity** is one of the most common metrics for evaluating language models. It is defined as the exponentiated average negative log-likelihood of a sequence, in our case, the sequence of generated tokens. The perplexity measures the model's ability to predict uniformly among the set of specified tokens in a corpus. A low perplexity indicates a good model. The final perplexity is obtained by averaging the perplexity of each generated text in the test set and we do not consider tokens occurring in the prompt during the computation.

The metrics are calculated for each model: *Llama3.1*, *Llama3.1-NA-100k* and *llama3-NA*. Results of the evaluation are reported in Table 1.

[7]NVIDIA RTX A6000 with 48GB of VRAM
[8]https://unsloth.ai/
[9]The model adapter is available on Hugging Face: https://huggingface.co/swap-uniba/llama3-it-pa-100k-adapter.
[10]The model adapter is available on Hugging Face: https://huggingface.co/swap-uniba/llama3-it-pa-300k-adapter.

Prompt length	20			40		
Model	Llama3.1	Llama3.1-NA-100k	Llama3.1-NA	Llama3.1	Llama3.1-NA-100k	Llama3.1-NA
BLEU-1	.132	.142	.152	.170	.193	.182
BLEU-2	.071	.102	.110	.111	.148	.142
Rouge-1	.313	.436	.448	.402	.509	.521
Rouge-2	.163	.294	.304	.266	.384	.397
Rouge-L	.254	.377	.387	.342	.451	.462
BERTscore	.705	.772	.777	.744	.796	.802
Perplexity	2.180	1.557	1.513	2.760	1.586	1.539

Table 1
Evaluation results considering different prompts length and training corpus size.

Results show that fine-tuned models always overcome the base model for all metrics. We observe a slight decrease in BLUE when the training corpus size increases. However, the differences between *Llama3.1-NA-100k* and *Llama3.1-NA* are minimal. Also, the perplexity has the same behaviour. This is an interesting outcome since we can adapt the model using a moderate number of documents. Moreover, we observe that the increase in performance is more evident when the prompt length is equal to 20. This behaviour is obvious since the text generated with a prompt length of 40 has more tokens in common with the original text. Nevertheless, the results prove that the tuned models can also increase the generation performance with a short prompt.

It is essential to highlight that we only measure the coherence of the generated text against the reference text in the test set. We do not check if the generated text is correct and contains accurate information. This is out of the scope of our work. To use the model in a real scenario, it is necessary to instruct the model on specific tasks through instruction tuning. The scope of our work is to provide a language model that can generate text more coherently with a new domain. All the results prove the effectiveness of our methodology.

Table 2 reports the results of each model on a set of standard benchmarks used to evaluate the ability of LLMs to solve several tasks. We consider the same set of benchmarks adopted by the Open Italian LLM leaderboard[11]. The involved benchmarks are:

- **HellaSWAG** is a dataset for studying grounded commonsense inference. It consists of 70k multiple-choice questions about grounded situations. Each question has four answer choices.
- The AI2's Reasoning Challenge (**ARC**) dataset is a multiple-choice question-answering dataset containing questions from science exams from grade 3 to grade 9.
- **MMLU** (Massive Multitask Language Understanding) is a benchmark designed to measure knowledge acquired during pretraining by evaluating models exclusively in zero-shot and few-shot settings. The benchmark covers 57 subjects across STEM, the humanities, the social sciences, and more.

The Italian benchmark relies on a machine-translated version of the dataset above.

The column Δ reports the difference in performance with respect to the original model Llama3.1. We observed a performance decrease as expected since we finetuned the model on new data and a different domain. However, if we consider both results in Table 1 and 2, we can conclude that the best choice is to fine-tune the model on 100k documents since the generation performance on the test set is good and the difference with the original model is about -8.5%. We plan to test the model on specific tasks related to the legal domain to understand better if the fine-tuning can improve both the generation and abilities of the model to solve domain-specific problems.

[11] https://huggingface.co/spaces/mii-llm/open_ita_llm_leaderboard

Model	hellaswag_it	arc_it	mmlu_it	avg	Δ (%)
Llama3.1	0.6256	0.4559	0.5593	0.5469	-
Llama3.1-NA-100k	0.5919	0.4166	0.4924	0.5003	8.53
Llama3.1-NA	0.5505	0.3807	0.4549	0.4620	15.52

Table 2
Performance of each model according to the Open Italian LLM leaderboard.

5. Conclusions and Future Work

This work proposes a methodology for adapting an open LLM to the Italian legal domain. To achieve this goal, we build a corpus of legal documents extracted from the Normattiva website. We also create an ad hoc scraper to ensure high-quality extracted text. Then, the corpus is exploited to adapt the `Llama-3.1-8b` model using continuous pre-training and LoRA. We also investigate different training corpus sizes.

We measure the adapted models' ability to complete sentences coherently according to the new domain to evaluate the effectiveness. Results prove that adapted models overcome the original model in all metrics, considering different prompt lengths. In future work, we plan to extend the analysis to other open LLMs and test fine-tuned adapted models on specific legal tasks.

Acknowledgments

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI (CUP H97G22000210007) under the NRRP MUR program funded by the NextGenerationEU.

References

[1] G. Bonetta, C. D. Hromei, L. Siciliani, M. A. Stranisci, Preface to the Eighth Workshop on Natural Language for Artificial Intelligence (NL4AI), in: Proceedings of the Eighth Workshop on Natural Language for Artificial Intelligence (NL4AI 2024) co-located with 23th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2024), 2024.

[2] S. Li, L. Yi, A few-shot entity relation extraction method in the legal domain based on large language models, in: Proceedings of the 2024 Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence, DEAI '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 580–586. URL: https://doi.org/10.1145/3675417.3675513. doi:10.1145/3675417.3675513.

[3] J.-H. Huang, C.-C. Yang, Y. Shen, A. M. Pacces, E. Kanoulas, Optimizing numerical estimation and operational efficiency in the legal domain through large language models, 2024. URL: https://arxiv.org/abs/2407.19041. arXiv:2407.19041.

[4] J. Savelka, Unlocking practical applications in legal domain: Evaluation of gpt for zero-shot semantic annotation of legal texts, in: Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law, ICAIL '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 447–451. URL: https://doi.org/10.1145/3594536.3595161. doi:10.1145/3594536.3595161.

[5] A. Al Zubaer, M. Granitzer, J. Mitrović, Performance analysis of large language models in the domain of legal argument mining, Frontiers in Artificial Intelligence 6 (2023). URL: https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2023.1278796. doi:10.3389/frai.2023.1278796.

[6] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, Legal-bert: The muppets straight out of law school, 2020. URL: https://arxiv.org/abs/2010.02559. arXiv:2010.02559.

[7] D. Licari, G. Comandè, Italian-legal-bert: A pre-trained transformer language model for italian law., in: EKAW (Companion), 2022.

[8] 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, arXiv preprint arXiv:2106.09685 (2021).
[9] T. Le Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, et al., Bloom: A 176b-parameter open-access multilingual language model, hal-03850124f (2023).
[10] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288 (2023).
[11] A. Dubey, et al., The llama 3 herd of models, 2024. URL: https://arxiv.org/abs/2407.21783. arXiv:2407.21783.
[12] P. Basile, E. Musacchio, M. Polignano, L. Siciliani, G. Fiameni, G. Semeraro, Llamantino: Llama 2 models for effective text generation in italian language, arXiv preprint arXiv:2312.09993 (2023).
[13] M. Polignano, P. Basile, G. Semeraro, Advanced natural-based interaction for the italian language: Llamantino-3-anita, arXiv preprint arXiv:2405.07101 (2024).
[14] P. Basile, L. Siciliani, E. Musacchio, M. Polignano, G. Semeraro, Adapting bloom to a new language: A case study for the italian, IJCoL. Italian Journal of Computational Linguistics 10 (2024).
[15] Z. Hu, Y. Lan, L. Wang, W. Xu, E.-P. Lim, R. K.-W. Lee, L. Bing, S. Poria, Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models, arXiv preprint arXiv:2304.01933 (2023).