# Experimenting Task-Specific LLMs

Stefano Scotta[1], Alberto Messina[1]

[1]RAI - Centro Ricerche, Innovazione Tecnologica e Sperimentazione, Via Giovanni Carlo Cavalli 6, 10138, Turin, Italy

**Abstract**

In this work, we present an example of how a relatively small Large Language Model (LLM) fine-tuned to perform a simple and well defined task (assigning titles to news articles) could perform similarly or even better than huge LLMs which are created to respond to any question. This approach of specializing smaller LLMs on simpler tasks is also interesting because it goes in the direction of making this technology more sustainable and available to a higher number of entities that usually could not use these expensive models, both for economic and data policy reasons. We also present a couple of examples of how can be evaluated the performances of LLMs when the task is specified as in the example that we present in this work.

**Keywords**

LLM, LoRA, LLama 2, fine-tuning, PEFT, Italian, news titles, benchmark

## 1. Introduction

The modern wave of applications based on artificial intelligence is characterised by the widespread adoption of large language models (LLM) [2], which - due to their undisputed ability to shorten the gap between technology and its exploitation in use cases relevant for the business - represent today the cornerstone of almost all attempts at implementing efficient and hugely flexible text processing pipelines. Despite this great flexibility, experiments often show how in specific use cases traditional approaches may show a better balance between performance and footprint. Clearly, huge LLMs like the new GPT models, which are engineered, through an extensive process of reinforcement learning from human feedback (see [3]), to be able to answer to any instruction in an *optimal* way, are most powerful in general, but this power comes at non-trivial costs in terms of e.g. payment for computing services as well as environmental costs. These costs may soon become not affordable both for business entities and for society at large.

On the other hand, LLMs' potential to introduce creative elements in some of these "simpler" cases is certainly worth being explored if we are ready to accept some additional requirements in terms of needed resources.

The main idea behind this work and other similar experiments we are conducting is to use an LLM for one of such specific tasks and to demonstrate that, in this task, it can be more useful to precisely fine-tuning a smaller model than using a giantic one (an interesting reading about that is [4]). The underlying hypothesis is that whenever the variability of the input that the

*NL4AI 2023: Seventh Workshop on Natural Language for Artificial Intelligence, November 6-7th, 2023, Rome, Italy [1]*

stefano.scotta@rai.it (S. Scotta); alberto.messina@rai.it (A. Messina)

0000-0003-1078-2985 (S. Scotta); 0000-0002-8262-2449 (A. Messina)



CEUR Workshop Proceedings (CEUR-WS.org)

LLM receives is reduced, also the quantity of possible outputs is reduced, which means that the complexity - and therefore the dimension of the model - does not need to be huge, in favour of a better balance between performance and resource requirements.

To start testing this hypothesis experimentally we decided to focus on a specific task which has a business relevance in the media production domain, i.e. that of assigning a title to a news article. This choice is done also because it is known that LLMs are less eager to hallucinate when they have to analyze, re-write or summarize a given text and this is due to their probabilistic "knowledge" of language (related to this we suggest [5]), so the task chosen is a typical task for which an LLM can be very useful.

Other recent and relevant works in this area notable include [6], an end to end framework for assigning headlines to news stories, of which this work is equivalent w.r.t. the single-document encoder-decoder block only, which however has been specifically trained on a very large corpus beforehand. In contrast, the principal goal of this work is to show that readily available language models, after some relatively small fine-tuning, could perform similarly or better than the biggest LLMs available, when we use them for a specific task. Comparing our approach to [6] could be still part of some future extension of this research.

## 2. Model and fine-tuning technique

### 2.1. Base model

The model we used as a solid base to optimize on the specific task of assigning titles is Llama 2 [7], the evolution of the first LLM released by Meta [8]. In particular, we used the 7 billions parameters version (the smallest one).

Moreover, because of the good performances on generic tasks of instruct-tuned LLMs like [9], which are a result of a fine-tuning process on generic couples instruction/answer (see the main idea on [10]), instead of starting from the native Llama 2, we decided to use an already tuned version of it, namely the model Nous-Hermes-Llama2-7b (Hermes7b hereinafter) available at [11]. This model is fine-tuned from Llama 2 on a dataset of instructions/answers with the following structure:

```
### Instruction:
<instruction/question to be answered>
### Response:
<answer to the instruction>
```

Since this model performs well on English instructions and we are interested in assigning titles to Italian news articles, we developed an "Italian version" of it. So, following the same strategy used to obtain Camoscio [12] (where the authors fine-tuned Llama [8]), we used this as the base model, since it is already fine-tuned to follow the scheme above of answering a given instruction, and fine-tune it with a dataset composed of entries structured in the same way but written in Italian. In particular, we fine-tuned it with 120k random entries of the dataset [13] using LoRA approach [14] (as the authors did in [12]) obtaining the LoRA adapters that, merged with the original weights of [11], constitute the model Hermes7b-ITA [15], see Section 2.3 for more details. We do not use directly Camoscio, the model developed in [12] based on

[8], mainly because of two reasons: first, Llama 2 as base model has a higher context length with respect to Llama (4096 tokens against 2048), is trained on more data (2T tokens against 1T) and performs better on various benchmarks (see [7]); furthermore, the dataset we used ([13]) is bigger than the one used for Camoscio and generated with more advanced models (GPT 4 and GPT 3.5 against GPT 3).

In summary, Hermes7b-ITA is the result of a double fine-tuning process on the base model Llama 2: the first done by the group Nous Research to make it optimized to answer generic instructions and the second, done with the LoRA approach, to further fine-tune the resulting model ([11]) to do the same but in Italian. The resulting model can now be used to answer to generic instructions and, since it is the whole model with the adapters merged with the original weights, can be further fine-tuned on more specific tasks (in particular in Italian).

## 2.2. Fine-tuned model

We finally used the base model Hermes7b-ITA as base model to perform the specific task of assigning a title to the text of a news article. To do so we use again LoRA approach to fine-tune the model on couples text/title structured in the same way as the instruction used to fine-tune Hermes7b and then Hermes7b-ITA:

```
### Instruction: Assegna un titolo al seguente articolo giornalistico.

<text of the news article>

### Response:
<title of the news article>
```

Notice that the first part of the instruction is constant (in English it means "Assign a title to the following journalistic article."), so that the only variability during the fine-tuning process is in the content and title of the news article considered, while the task is fixed.

The resulting model, newsTitler hereinafter, is an extremely specialized LLM, able to assign a representative title to a news passed to it according to the above prompt . The output is simply the title without any other "textual noise" (like "A good title could be <title>") typical of LLMs.

We would like to remark that other LLMs, as Hermes7b and Hermes7b-ITA or even the OpenAI GPT models (3.5 and 4) are already able to answer to a similar instruction zero-shot, however we show that 1) our fine-tuned LLM performs better than the base models; and 2) similarly to the state-of-the-art GPT models. As anticipated, GPT 4 OpenAI's model ([16]) performs really well also on this task but, as we will see in Section 3, it does not perform better than newsTitler. Regarding the other models: 1) Hermes7b, trained/fine-tuned in English, has the problem that often the answer is in English; 2) Hermes7b-ITA does not perform too bad but beside adding "text noise" to the title in output, often simply answers by repeating the first words of the article or giving a title not strictly related to the content of the news.

## 2.3. Datasets and technical details

The first fine-tuning process (i.e. from Hermes to Hermes7b-ITA [15] ) is based on LoRA approach with the following hyerparameters and LoRA configuration: train epochs = 3, learn-

ing rate = 2e-4, mixed precision training = float16, LoRA r = 8, LoRA alpha=16, target modules=[‘q_proj’,‘v_proj’], LoRA dropout=0.05, bias=‘none’, task type=TaskType.CAUSAL_LM. The dataset consists of around 120k random entries of the dataset [13] formatted according to the prompt in section 2.1. The whole fine-tuning procedure lasted around 78 hours on a single GPU NVIDIA A100 40Gb.

The second fine-tuning process (i.e. from Hermes7b-ITA to newsTitler) is as well based on the LoRA approach with the same hyperparameters of the one described above. The dataset in this case consists of couples of around 20k of news titles and text, published by Rai in the period 01/01/2022 – 09/03/2023, formatted according to the template in Section 2.2. The urls of these news article can be found at [17] in the file *urls_train_set.csv*. The whole fine-tuning procedure lasted around 24 hours on a single GPU NVIDIA A100 40Gb.

To do the benchmarks in Section 3 we used the same prompt in section 2.2 (obviously omitting the title in order to be generated by the models) to assign titles to a set of 1148 news articles published by Rai in the period 10/03/2023 - 04/05/2023, whose urls are available at [17].

## 3. Benchmark and comparison between models

A common challenge about LLMs is the evaluation of the quality of their performance due to the huge quantity and variety of the tasks for which they are employed. It is difficult to say that an LLM is “better” than another, even if various tries are being done, see for example [18, 19].

In the case of specific tasks, like the one we consider in this work, the number of variables decreases substantially. Indeed, in this case the task is always the same and the output can be considered “better” or “worse” depending on how close it is to the real title assigned by the journalist, making the assumption that the latter can be considered as ground truth. Under these assumptions we present below different metrics we used to compare the outputs of Hermes7b, Hermes7b-ITA, newsTitler and GPT 4.

The prompt used to generate titles with Hermes7b, Hermes7b-ITA and newsTitler is the same as the one in Section 2.2. To generate the titles with GPT 4 we used the following, slightly different, prompt in order to avoid *text noise* like “Un buon titolo potrebbe essere: <title>” (Italian for “A good title could be <title>”)

```
Analizza il contenuto del seguente testo e fornisci un titolo
rappresentativo (riporta esclusivamente il titolo):
```

and then we analyzed the titles generated to eliminate the aforementioned *noise* and make the comparison between models more significant.

Note that, as pointed out firstly in [20], LLMs can improve their performances strongly when they are not prompted in a “zero-shot” way, as above, but in a “few-shot” setting, and so we would expect also in our case if GPT-4 was prompted with few-shot. However, the variety of typical news articles would require a strategy to carefully select the example(s) to insert in the few-shot prompt: an article about sport probably should be titled following an example of the same type in the prompt. The development of this kind of strategy, which could benefit from recent developments like [21], goes beyond the scope of this paper which is principally aimed to compare an open source model with GPT-4 in the simplest possible setting, which is the zero-shot case.

### 3.1. ROUGE and BLEU scores

In this Section we consider as metrics the ROUGE ([22]) and the BLEU ([23]) scores, widely used to compare an automatic produced text with a reference one. In this case we use these scores to compare the titles given by the journalists with the titles produced by the LLMs in the test set. Let us briefly recall what are these two metrics and which implementations we used:

- The ROUGE score is a recall measure (going from 0 to 1), it depends on the fraction of words/groups of words in the generated title that are present in the reference one. In particular, we considered the ROUGE-L score which depends on the longest common subsequence of words appearing in both titles (see [22] for details);
- the BLEU score is a precision measure, indeed it depends essentially on the fraction of words/groups of words in the generated output that are present in the reference one. To calculate this score we use the implementation in [24] normalized so that the maximum value is 1 and the minimum is 0.

These are two very simple metrics which do not take really in account the meaning of the sentences, but reasonably useful to compare the mean value of the scores on all the titles generated by each model. The results are summarized in Table 1.

**Table 1**
Mean and standard deviation of the ROUGE and BLEU scores calculated on the titles for the news in the test set generated by the LLMs considered.

| | GPT 4 | Hermes7b | Hermes7b-ITA | newsTitler |
|---|---|---|---|---|
| mean ROUGE | 0.250 | 0.096 | 0.249 | **0.310** |
| std ROUGE | 0.149 | 0.100 | 0.158 | 0.181 |
| mean BLEU | 0.090 | 0.028 | 0.088 | **0.116** |
| std BLEU | 0.090 | 0.036 | 0.095 | 0.118 |

According to these two metrics the model which performs the best is newsTitler, but it is interesting to notice that GPT 4 does not perform significantly better than Hermes7b-ITA although this is quite smaller compared to the OpenAI's model and can run on a single GPU locally. Moreover, as it was predictable, Hermes - which does not have substantial training or fine tuning in Italian - performs significantly worse than the other three.

### 3.2. Cosine similarity between embeddings

The previous metrics have a big limitation: they only depend on the words and not on the meaning of them or of their combination. So, in order to better capture the meaning of the titles proposed and compare them with the meaning of the real titles, we need first to convert each title in some mathematical object taking in account text semantics. The most used way to do this is to evaluate some kind of embedding for each title, see [25] for example, obtaining a kind of "translation" of the meaning of a sentence in a mathematical array.

The embeddings that we chose are calculated using the model sentence-transformers/all-MiniLM-L6-v2, based on the results in [25] and [26], which converts any sentence in an array of 384 real-valued elements.

So, in order to compare two titles we calculate the embedding for each of them and we use some function to account for their similarity, like the cosine similarity.

Following the above procedure, for each article in the test set we evaluated the embeddings for the generated titles and for the real one and, lastly, we compute the cosine similarity between them. In Table 2 we show the mean of the cosine similarities between the titles generated by each LLM and the real titles in the test set. The results are consistent with the BLEU and ROUGE scores in Section 3.1, indeed also according to this evaluation we see that the model that performs best is newsTitler with GPT 4 and Hermes7b-ITA having similar results.

**Table 2**
Mean and standard deviation of the cosine similarity between the embeddings of the titles generated by the LLMs considered and the ones of the original titles.

| | GPT 4 | Hermes7b | Hermes7b-ITA | newsTitler |
|---|---|---|---|---|
| mean | 0.640 | 0.341 | 0.621 | **0.656** |
| std | 0.136 | 0.177 | 0.145 | 0.143 |

## 4. Conclusion and future works

In this work we described a first experiment aimed at empirically showing that it is not always necessary to rely on huge, expensive and often proprietary LLMs that, simply prompted in the opportune way are (or seem to be) able to respond to any kind of question. Indeed, we show that, if the task for which we think to use an LLM is well defined and confined, it is possible to use smaller open source models reaching results comparable or even better than huge LLMs (on the specific task). We considered the example of assigning titles to news articles, so a very simple but business-relevant task, but considering a more complex task this approach can be easily extended. We argue that if the task could be decomposed in many smaller tasks it could be better and cheaper to fine tune a series of small LLMs to do each of the smaller tasks, and even in this case (unless the micro tasks are hundreds) this approach would be overall more convenient than using a huge LLM to do everything.

As part of future work in this area we would like to extend experimentation including different kind of tasks, possibly a combination of them, as hypothesized above. Moreover, it would be interesting to develop new benchmarking methods (task dependent, clearly) in order to assess, for each job, which model (to be intended as result of a base model plus fine-tuning process) would be the best each time.

## Acknowledgments

This work was partially supported by European Union's Horizon 2020 research and innovation programme under grant number 951911 - AI4Media.

## References


[1] E. Bassignana, D. Brunato, M. Polignano, A. Ramponi, Preface to the Seventh Workshop on Natural Language for Artificial Intelligence (NL4AI), in: Proceedings of the Seventh Workshop on Natural Language for Artificial Intelligence (NL4AI 2023) co-located with 22th International Conference of the Italian Association for Artificial Intelligence (AI* IA 2023), 2023.

[2] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al., A survey of large language models, arXiv preprint arXiv:2303.18223 (2023).

[3] OpenAI, Introducing chatgpt, 2022. URL: https://openai.com/blog/chatgpt.

[4] M. Honnibal, Against llm maximalism, https://explosion.ai/blog/against-llm-maximalism, 2023.

[5] S. Wolfram, What is chatgpt doing ... and why does it work?, https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work, 2023.

[6] X. Gu, Y. Mao, J. Han, J. Liu, Y. Wu, C. Yu, D. Finnie, H. Yu, J. Zhai, N. Zukoski, Generating representative headlines for news stories, in: Proceedings of The Web Conference 2020, 2020, pp. 1773–1784.

[7] 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).

[8] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[9] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T. B. Hashimoto, Stanford alpaca: An instruction-following llama model, https://github.com/tatsu-lab/stanford_alpaca, 2023.

[10] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, H. Hajishirzi, Self-instruct: Aligning language model with self generated instructions, arXiv preprint arXiv:2212.10560 (2022).

[11] Nous Research, Nous-hermes-llama2-7b, https://huggingface.co/NousResearch/Nous-Hermes-llama-2-7b, 2023.

[12] A. Santilli, E. Rodolà, Camoscio: an italian instruction-tuned llama, arXiv preprint arXiv:2307.16456 (2023).

[13] Rai - CRITS, Orca ITA 200k, https://huggingface.co/datasets/raicrits/Orca_ITA_200k, 2023.

[14] 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).

[15] Rai - CRITS, Hermes7b ITA, https://huggingface.co/raicrits/Hermes7b_ITA, 2023.

[16] OpenAI, Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).

[17] Rai - CRITS, news urls, https://huggingface.co/datasets/raicrits/news_urls, 2023.

[18] Y. Chang, X. Wang, J. Wang, Y. Wu, K. Zhu, H. Chen, L. Yang, X. Yi, C. Wang, Y. Wang, et al., A survey on evaluation of large language models, arXiv preprint arXiv:2307.03109 (2023).

[19] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al., Judging llm-as-a-judge with mt-bench and chatbot arena, arXiv preprint

arXiv:2306.05685 (2023).
[20] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.
[21] O. Khattab, A. Singhvi, P. Maheshwari, Z. Zhang, K. Santhanam, S. Vardhamanan, S. Haq, A. Sharma, T. T. Joshi, H. Moazam, H. Miller, M. Zaharia, C. Potts, Dspy: Compiling declarative language model calls into self-improving pipelines, 2023. arXiv:2310.03714.
[22] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, 2004, pp. 74–81.
[23] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 311–318.
[24] M. Post, A call for clarity in reporting BLEU scores, in: Proceedings of the Third Conference on Machine Translation: Research Papers, Association for Computational Linguistics, Belgium, Brussels, 2018, pp. 186–191. URL: https://www.aclweb.org/anthology/W18-6319.
[25] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: http://arxiv.org/abs/1908.10084.
[26] N. Reimers, I. Gurevych, Making monolingual sentence embeddings multilingual using knowledge distillation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2020. URL: https://arxiv.org/abs/2004.09813.