Scaling Effects of Instruction Tuning in Encoder-Based Language Models

Anonymous ACL submission

Abstract

Instruction tuning has significantly improved the task-following capabilities of decoder-based language models, yet its effects on encoder-based architectures remain underexplored. This study investigates instruction tuning in the XLM-R model family for prompted classification tasks, analyzing models ranging from $250M$ to $10B$ parameters under three training paradigms: standard fine-tuned, prompted base models, and instruction-tuned prompted models. Our experiments, conducted on a subset of SuperGLUE classification datasets, show that instruction tuning significantly benefits larger XLM-R variants, particularly those with at least 500M parameters. However, the performance gains do not scale directly with model size. Notably, XLM-R_{large} achieves competitive improvements, while XLM-R_{XL} underperforms despite its substantially larger parameter count. These findings suggest that pre-training data quality and quantity may play a key role in how well encoder-based models leverage instruction tuning. Additionally, we observe that the alignment between instruction tuning data and downstream tasks influences performance, underscoring the importance of data diversity. Our findings contribute to a more nuanced understanding of instruction tuning in encoder models and offer insights into optimizing their task-following capabilities.

1 Introduction

In recent years, decoder-based large language models (LLMs), such as GPT (OpenAI, 2024) and Gemini (et al., 2024), have dominated the field of natural language processing (NLP). Their impressive task-following capabilities, particularly after instruction tuning, have positioned them as the preferred choice for various applications (Kumar, 2024; Yang et al., 2024; Zhang et al., 2024). However, the research focus on decoder-based architectures has inadvertently led to the under-exploration of instruction tuning in encoder-based models like BERT (Devlin et al., 2019a) and its derivatives (Lin et al., 2022).

Recent studies (Xiao et al., 2024), suggest that instruction tuning can indeed benefit encoder models models, yet systematic and comprehensive investigations remain scarce. Addressing this research gap is crucial, as understanding the impact of instruction tuning on encoder models could enable the development of more data-efficient and adaptable NLP systems.

This study aims to investigate the impact of instruction tuning on the XLM-R models (Conneau et al., 2020) of different sizes in a prompted classification setting. Specifically, we seek to determine whether instruction tuning enhances the model's ability to follow task instructions and improve classification performance when primed with a task-specific prompt. Given the demonstrated success of instruction tuning in decoder-based models (Ouyang et al., 2022), we hypothesize that similar benefits may extend to encoder-based models.

To systematically explore this question, we define several key objectives. First, we aim to evaluate whether instruction tuning leads to measurable improvements in classification accuracy. Specifically, we examine its impact on the XLM-R model family when classification is modeled as a cloze task (Trinh and Le, 2019) with problem-specific prompting. Additionally, we analyze the role of model size in shaping the effectiveness of instruction tuning, comparing performance across different XLM-R variants, such as base, large, Xl and XXl. Understanding these scaling effects is crucial, as it could provide insights into whether instruction tuning benefits are dependent on model capacity. Another important goal is to assess whether instruction tuning reduces the need for labeled data in prompted classification tasks, making encoder models more data-efficient.

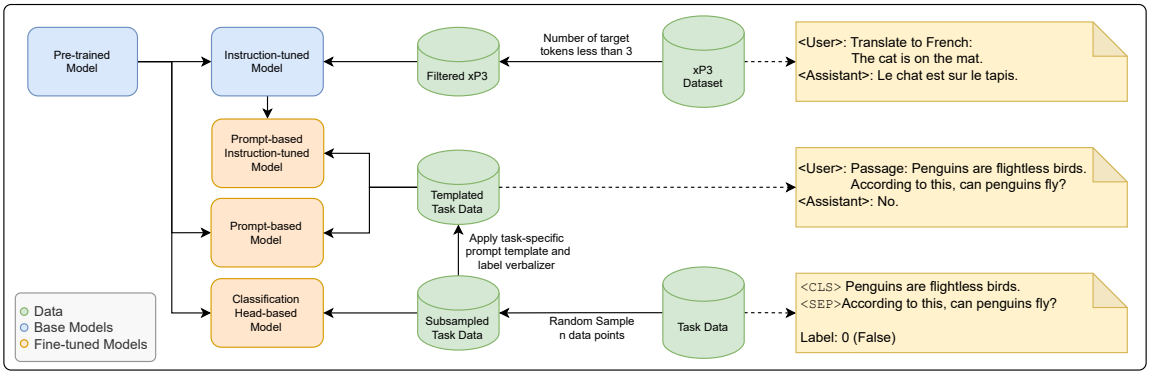

Figure 1: High-level illustration of the data preparation and model training processes.

To address these research objectives, we fine-tuned XLM-R models of varying sizes on an instruction-tuning dataset before applying them to classification tasks. This fine-tuning process enabled the models to learn from diverse task formulations presented as natural language instructions. We then evaluated the models in a prompted classification setup, leveraging the methodology from (Scao and Rush, 2021) to systematically measure data efficiency across different model sizes. The results were analyzed to determine whether instruction tuning improves task performance and whether its effectiveness scales with model size.

This study's main contribution is the systematic exploration of instruction tuning for encoder-based models in prompted classification. We investigate whether instruction tuning enhances their ability to follow natural language prompts and examine model size as a key factor, analyzing the scaling effects on instruction-tuned encoder models, inspired by trends observed in decoder-based models (Wu and Tang, 2024).

The paper is structured as follows: In the next section we present the most relevant papers related to this work. In Section 3 we present our proposed approach. Section 4 details the experimental setup of this study. In Section 5 we present the results and discuss the findings. Finally, in Section 6 we present our conclusions and delineate future work.

2 Related Work

The empirical benefits of task-specific prompting in fine-tuning pre-trained language models have been systematically explored in the literature. Scao and Rush (2021) investigate the effectiveness of prompting compared to traditional head-based fine-tuning for text classification across multiple tasks and dataset sizes. Their findings, based on experiments with a single model size, highlight that prompting significantly enhances sample efficiency, offering improvements equivalent to hundreds of data points. These results reinforce the utility of prompting strategies in improving the efficiency of supervised learning with transformer-based architectures. A recent study (Clavié et al., 2025) examines MLM-based generative classification, demonstrating that *ModernBERT-Instruct* can replace traditional classification heads and achieve competitive zero-shot performance against LLMs. Unlike the former study, which primarily examines the potential of masked language modeling in a classification setting using a model of fixed size, *ModernBERT* (Warner et al., 2024a), and focuses on comparisons with decoder models, our approach systematically evaluates instruction tuning across multiple XLM-R model sizes.

Instruction tuning has emerged as a key paradigm for enhancing the generalization capabilities of LLMs. This approach, wherein models are fine-tuned on diverse NLP tasks framed as natural language instructions, has demonstrated considerable success, particularly in decoder-based architectures such as GPT and T5 (Ouyang et al., 2022). The ability of instruction-tuned models to follow prompts and adapt to a wide range of tasks has been linked to critical scaling factors, including model size, training data volume, and computational resources (Kaplan et al., 2020; Ouyang et al., 2022). Larger models, in particular, exhibit more pronounced benefits from instruction tuning, suggesting that scale plays a fundamental role in emergent instruction-following capabilities (Tay et al., 2023).

While instruction tuning has primarily been explored in decoder-based models, recent efforts have

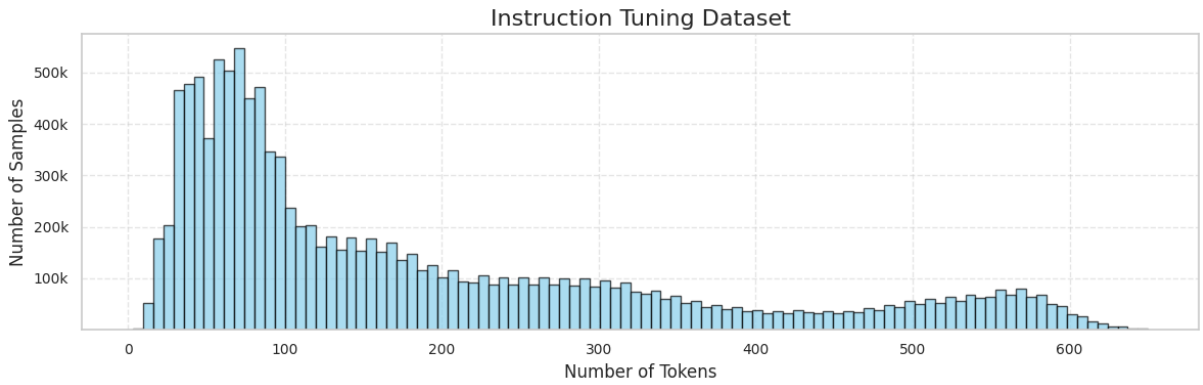

Figure 2: Token count distribution of samples in the filtered *xP3* instruction tuning dataset.

sought to adapt this approach to encoder-based architectures. Xiao et al. (2024) introduce Instruct-XLMR, a fine-tuned version of the XLM-R_{XL} and XLM-R_{XXL} models (Conneau et al., 2020), demonstrating its competitive performance against state-of-the-art instruction-tuned models such as BLOOMZ and mT0 (Muennighoff et al., 2023). Their study provides a critical analysis of the strengths and limitations of instruction tuning in encoder models. This work presents some key insights that include: Improved efficiency and inference speed compared to autoregressive models, enhanced performance on instruction-following tasks, despite reduced capabilities in long-text generation and few-shot learning and the potential for further optimization in adapting encoder-based models to instruction tuning.

Despite these advancements, the application of instruction tuning to encoder models remains an open research area. While preliminary results suggest that BERT-based architectures can be viable instruction followers, systematic investigations into the role of model size, dataset diversity, and optimization strategies have not been fully and thoroughly addressed. Addressing these gaps is crucial for broadening the applicability of instruction tuning beyond decoder-based models and unlocking new capabilities in efficient NLP model adaptation.

3 Proposed Methodology

This work explores the impact of instruction tuning on BERT-like models for classification tasks within a prompting-based setting. To this end, we compare three distinct training paradigms:

- **Standard Fine-Tuned Models:** These models are trained end-to-end with a classification head, representing the traditional fine-tuning approach without any prompting.
- **Prompted Base Models:** Pre-trained models fine-tuned and evaluated using a fixed task-specific prompt without additional instruction tuning.
- **Instruction-Tuned Models:** These models undergo instruction-tuning before task-specific fine-tuning while using the same prompt as the prompted base models.

A key research question in this study is how instruction tuning affects data efficiency and performance across different model sizes. The experimental setup follows established methodologies in prompt-based learning while systematically assessing the impact of instruction tuning (Scao and Rush, 2021). Figure 1 illustrates the data preparation and model training processes discussed in the next sections.

3.1 Instruction Tuning

We frame the instruction tuning process as a masked language modeling (MLM) task (Devlin et al., 2019b). Given an instruction-target pair from the instruction tuning dataset, we construct a chat-style text and train the model using MLM on the resulting token sequence. The masking strategy differs slightly from the standard MLM approach:

- A randomly selected 15% of all tokens in the sequence are masked, while special tokens and tokens belonging to the chat template are excluded from masking.
- The entire target sequence is always masked. During instruction tuning, all tokens in the target contribute to the loss, ensuring that

Model	#Params	#Tokens	#FLOPs	IT	FT
XLM-R_{base}	$250M$	N/A	N/A	$6h \times$ 2GPU	$6h \times$ 1GPU
XLM-R_{large}	$550M$	$6T$	2.2×10^{22}	$12h \times$ 2GPU	$10h \times$ 1GPU
XLM-R_{xl}	$3.5B$	$0.5T$	8.4×10^{21}	$16h \times$ 8GPU	$35h \times$ 2GPU
XLM-R_{xxl}	$10.7B$	$0.5T$	2.9×10^{22}	$48h \times$ 8GPU	$48h \times$ 8GPU

Table 1: Comparison of the XLM-R model family in terms of model size, pretraining resources, and compute investment for this study. The first three columns describe the model size and pretraining characteristics, where the number of tokens seen during pre-training are taken from Goyal et al. (2021) and the number of floating point operations are estimated (for details see appendix B). The last two columns show the compute resources used in this study: IT (Instruction Tuning) and FT (Fine-tuning), both measured in GPU hours.

the model learns to generate appropriate responses.

This approach ensures that both input and target tokens contribute to the learning signal. Given that prior studies have employed input token masking in similar settings, we adopt the same strategy for consistency (Scao and Rush, 2021; Clavié et al., 2025).

Following (Xiao et al., 2024), we use a subset of the *xP3* dataset (Muennighoff et al., 2023) for the instruction tuning process. To focus on tasks suited for classification, we apply a filtering criterion that retains only those samples where the target consists of at most three tokens. Additionally, we restrict our selection to the English split of the dataset, as the evaluation is also conducted solely on English benchmarks. The final instruction tuning dataset contains about 12.2 million samples and $2.7B$ tokens. Figure 2 shows the document length distribution in number of tokens.

3.2 Fine-tuning

The fine-tuning process for the three types of models follows distinct but related procedures, tailored to the specific approach each model uses.

Fully fine-tuned models are trained using a classification head, as originally proposed by Devlin et al. (2019b). These models are fine-tuned end-to-end, with the entire network, including both the pre-trained layers and the classification head, adjusted based on the downstream task.

For both the prompted base and the instruction-tuned models, the fine-tuning procedure is identical. The downstream task is modeled as a MLM problem, where the input to the model is a prompt following the template introduced during instruction tuning. This prompt includes a mask token at the position where the answer is expected. The model's task is to predict the correct token to replace the mask token. Other than during instruction tuning, no additional tokens are masked during the fine-tuning procedure.

Tasks are designed to be answerable with a single label-token, with output options restricted to a predefined set of label-tokens. In this work, these labels are typically limited to "yes" and "no", with one task additionally including "maybe". During fine-tuning, the model learns to predict the correct label-token based on the context provided in the prompt.

4 Experimental Setup

The objective of our experiments is to analyze the impact of instruction tuning on prompted BERT-like models for classification tasks. We systematically isolate three key variables: training data size, model size, and whether instruction tuning is applied. Other potential sources of variation, such as prompt format and verbalizer (label tokens), are held constant to ensure a controlled comparison.

4.1 Models and Scaling

To evaluate the scaling effects of instruction tuning, we conduct experiments on a range of models spanning different sizes. Specifically, we use the *XLM-R* model family (Conneau et al., 2020), including their Xl and XXl variants (Goyal et al., 2021). Table 1 provides an overview of the XLM-R model family, detailing their parameter counts and the number of tokens processed during pre-training. This allows us to assess how instruction tuning generalizes across different model sizes. These models cover a parameter range from approximately 250 million to 10 billion, enabling insights into whether larger models benefit disproportionately from instruction tuning.

4.2 Datasets and Evaluation Metrics

We select a subset of classification datasets from the SuperGLUE benchmark (Wang et al., 2020),

specifically *BoolQ*, *CommitmentBank* (*CB*), *Choice of Plausible Alternatives* (*COPA*), *Word-in-Context* (*WiC*), and *Recognizing Textual Entailment* (*RTE*). These datasets are chosen based on their diversity in reasoning types and their alignment with the capabilities of instruction-tuned language models:

- **BoolQ**: Binary classification task requiring models to answer yes/no questions based on a given passage.
- **CB**: Natural language inference task with three-way classification (entailment, contradiction, neutral), evaluated using F1-score.
- **COPA**: Causal reasoning task where the model must select the more plausible alternative.
- **WiC**: Lexical semantics task testing whether a given word is used with the same meaning in two different sentences.
- **RTE**: Recognizing textual entailment task where the model determines whether a premise entails a hypothesis.

These datasets encompass binary and multi-class classification problems, covering inference, factual knowledge, and lexical semantics, making them well-suited for evaluating the generalization effects of instruction tuning.

Evaluation follows standard metrics used in SuperGLUE: accuracy for most tasks, with the exception of *CB*, which is evaluated using macro-averaged F1-score due to its imbalanced label distribution. Additionally, we report the average performance increase as the ratio of the integrals of the performance curves, called *Integral Performance Ratio* (IPR). Let f and g be the performance curves, then the average performance increase is measured as:

$$\text{IPR}(f, g) = \frac{\int f(x)dx}{\int g(x)dx} - 1$$

where subtracting 1 ensures we capture the relative increase rather than the scaling factor. The IPR metric quantifies the percentage gain of f over g, providing a more holistic view of performance differences across the different training data sizes.

Due to the limitations of the *average prompt advantage* metric introduced by Scao and Rush (2021), as discussed in Appendix A, we opted for an alternative metric better suited to our analysis.

4.3 Training Data Regimes and Optimization

Our study focuses on the impact of instruction tuning in low-data settings. To systematically analyze its effect across different training data regimes, we evaluate models with varying amounts of training data, capping the number of examples at 128. This constraint ensures that all experiments are conducted under limited-data conditions, allowing us to assess the data efficiency of instruction tuning.

All models were instruction tuned for one epoch over the whole filtered *xP3* dataset using a global batch size of 128, a learning rate of 2×10^{-5}, and a weight decay of 0.01. We fine-tuned the models with a small learning rate of 1×10^{-5} and train for a high number of steps (> 300), as recommended in a prior work for stabilizing training in low-data regimes (Mosbach et al., 2021; Zhang et al., 2021). During fine-tuning, we maintained a fixed prompt format across tasks to control for prompt variability and ensure fair comparisons across training paradigms. The label set was constrained to "yes", "no", and "maybe" for the *CB* task. All fine-tuning experiments were repeated for three different random seeds, to ensure robustness of our findings.

For both instruction tuning and fine-tuning, we used FSDP (Zhao et al., 2023) for distributed training of larger models.

4.4 Computational Resources and Experiment Execution

All experiments were conducted on an NVIDIA DGX system equipped with 8 H100 (80GB) GPUs. The cumulative compute cost for the total of $1,260$ fine-tuning experiments amounted to 470 GPU hours. In addition to these experiments, an additional 612 GPU hours were allocated for instruction tuning of the models. Table 1 gives a more detailed overview of the invested GPU hours.

5 Experiments and Discussion

This section evaluates the impact of instruction tuning across different model sizes, analyzing its effectiveness in improving classification performance, examining the role of pre-training, fine-tuning and compute resources. Figure 3 illustrates the performance curves for three datasets, highlighting the behavior of the baseline, prompted, and instruction-tuned models. Additional results can be found in Appendix C. Overall, we observe that prompting generally improves performance over the baseline in most settings, supporting the findings of Scao

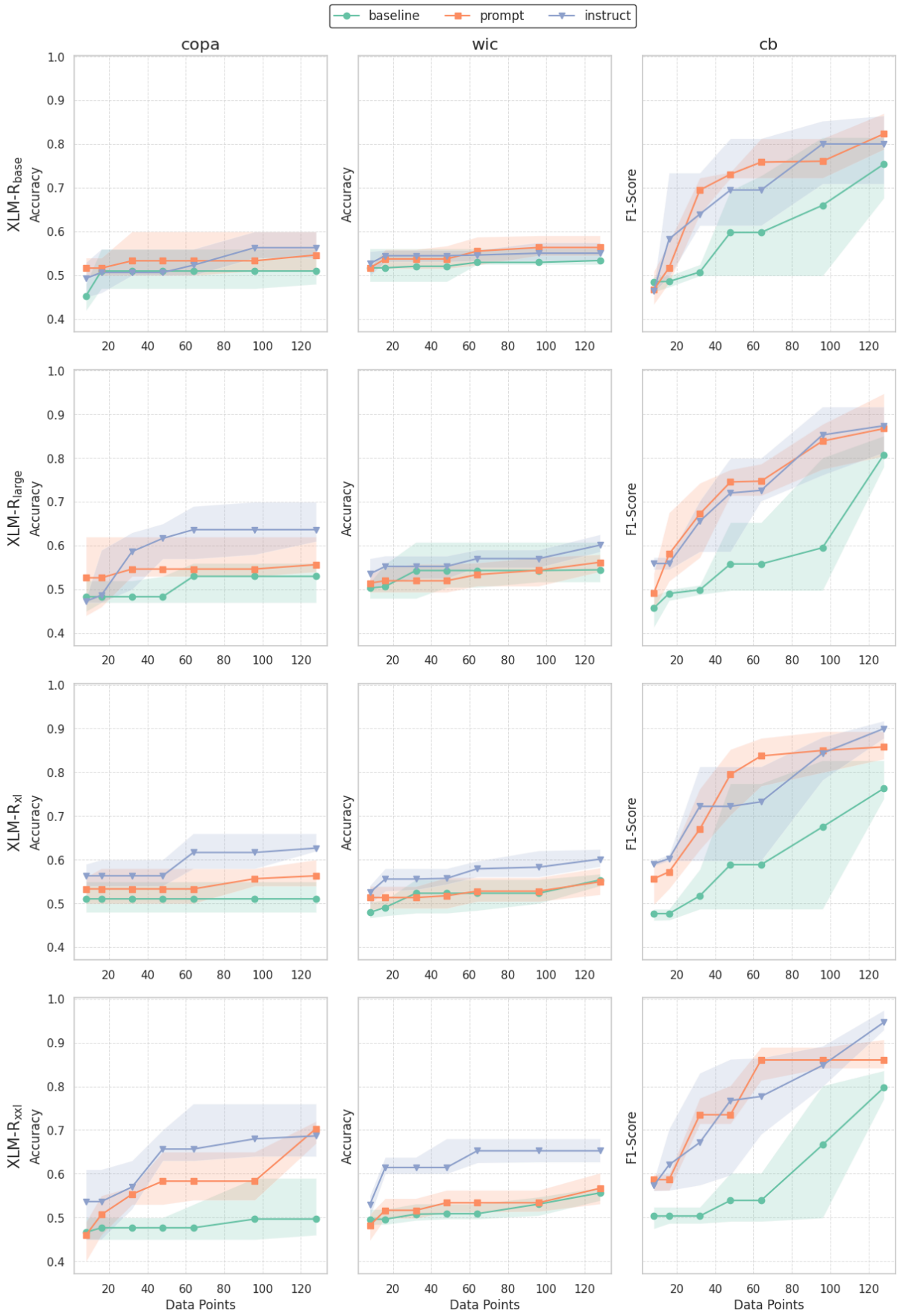

Figure 3: Performance curves of various XLM-R model sizes on the *COPA*, *WiC*, and *CB* benchmarks. Each plot presents the performance trends for the different fine-tuning strategies. The x-Axis indicates the amount of training data available for fine-tuning. The baseline corresponds to the standard head-based classification approach.

Model	BoolQ	CB	COPA	RTE	WiC	Average
XLM-R_{base}	3.0 ± 1.3	$\mathbf{1.5 \pm 2.0}$	-0.9 ± 7.5	4.0 ± 2.8	-0.9 ± 1.1	1.3 ± 3.8
XLM-R_{large}	7.3 ± 2.6	0.4 ± 7.4	10.6 ± 4.3	10.3 ± 1.8	7.6 ± 4.9	7.3 ± 5.5
XLM-R_{xl}	8.3 ± 2.1	-2.5 ± 2.6	8.3 ± 5.1	$\mathbf{12.3 \pm 3.0}$	5.9 ± 2.3	6.5 ± 5.8
XLM-R_{xxl}	$\mathbf{9.4 \pm 1.1}$	-0.2 ± 4.0	$\mathbf{11.6 \pm 4.7}$	8.4 ± 1.0	$\mathbf{20.0 \pm 2.9}$	$\mathbf{9.9 \pm 7.2}$

Table 2: Average Integral Performance Ratio (IPR) of instruction-tuned XLM-R models over their prompted base counterparts, reported in percentage points. A positive IPR signifies a relative improvement due to instruction tuning, while a negative IPR indicates a relative disadvantage.

and Rush (2021).

As stated previously, a key focus of our study is the impact of instruction tuning across different model sizes. For XLM-R_{base}, the results are mixed. While slight improvements are observed on *BoolQ* and *RTE*, the performance on other tasks remains comparable to prompting, suggesting that smaller models struggle to generalize effectively under instruction tuning. In contrast, XLM-R_{large} consistently benefits from instruction tuning, showing significant improvements across four out of five tasks. The trend continues with XLM-R_{XL}, which outperforms prompting but with slightly smaller gains. The performance of XLM-R_{XXL} further reinforces these findings, as it consistently outperforms the prompted model in nearly all tasks, with the strongest gains attributed to instruction tuning.

To quantify the improvements, we examine the Integral Performance Ratio (IPR) scores, summarized in Table 2. The scores align with the trends observed in Figure 3. Instruction-tuned models with more than $500M$ parameters achieve a substantial performance boost over their prompted counterparts, with an average improvement of 7.3%. The strongest gains are observed for XLM-R_{XXL}, particularly on the *WiC* task, where instruction tuning yields an increase of 20%. On average, XLM-R_{XXL} exhibits a 9.9% improvement across all tasks. XLM-R_{large}, despite being significantly smaller, also demonstrates strong performance gains. Meanwhile, XLM-R_{XL} benefits from instruction tuning but to a lesser extent compared to the large and XXL variants.

An analysis of dataset-specific trends reveals that the *CB* dataset exhibits the weakest improvements across all model sizes. Instruction-tuned models perform similarly to their prompted counterparts, in contrast to other datasets, where instruction tuning provides a clear advantage, particularly for models exceeding $500M$ parameters. We hypothesize that this discrepancy arises due to misalignment between *CB* and the instruction tuning data. One possible explanation is that data filtering during instruction tuning reduced task diversity, limiting the model's ability to generalize effectively. Figure 4 supports this hypothesis, showing that the instruction tuning dataset contains a high proportion of binary "yes"/"no" answers, whereas *CB* includes an additional "maybe" label. This label is significantly underrepresented, with only 5,829 samples, potentially leading to the observed performance gap. This suggests a major limitation of our study, as model performance is highly dependent on the choice of label tokens.

We initially expected instruction tuning benefits to scale consistently with model size. While XLM-R_{XXL} exhibits the largest gains, followed closely by XLM-R_{large}, the underperformance of XLM-R_{XL} compared to these two models is unexpected. One likely explanation lies in pre-training differences. All XLM-R models were trained on CC-100, consisting of $167B$ tokens, but for varying durations. As Table 1 indicates, the strong performance of the large model may be attributed to a higher number of tokens seen during pre-training, while both XL and XXL models appear undertrained. However, XLM-R_{XXL} compensates for this limitation due to its larger model size, enabling it to adapt better to instruction tuning despite insufficient pre-training. XLM-R_{XL}, however, does not recover as effectively. Additionally, XLM-R_{XL} has the lowest compute budget among the three largest models. XLM-R_{large} received 2.6 times the pre-training compute, and XLM-R_{XXL} approximately 3.5 times the compute budget of XLM-R_{XL}, further reinforcing the importance of pre-training investments for instruction tuning. These observations suggest that both pre-training diversity and compute resources play a crucial role in determining the effectiveness of instruction tuning.

Our findings align with prior work on the impact of pre-training on instruction following. Recent

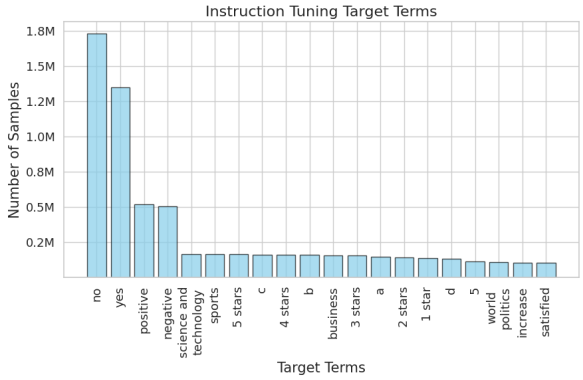

Figure 4: Frequency of the most common target terms in the filtered instruction tuning dataset.

studies highlight the significance of data quality, model size, dataset scale, and compute budget in determining instruction tuning efficacy (Gunasekar et al., 2023; Zhao et al., 2024). Our results are also consistent with the findings of Clavié et al. (2025), who demonstrated strong zero- and few-shot performance of *ModernBERT* (Warner et al., 2024b) when instruction tuned. This model was pre-trained on a modern text corpus of $2T$ tokens, representing a four-fold increase in training tokens and a roughly twelve-fold increase in data compared to the XLM-R_{XL} and XLM-R_{XXL} models. Although CC-100 is highly multilingual, encompassing over 100 languages, the dataset employed for *ModernBERT* focuses primarily on English. This monolingual dataset exhibits greater diversity in domains compared to *CC-100*. Furthermore, *ModernBERT*'s training data benefits from a more recent and sophisticated data collection and filtering methodology, reflecting current best practices. As data quality has been shown to be a critical factor in model performance for LLMs (Gunasekar et al., 2023; Zhao et al., 2024), this difference in data curation likely plays a substantial role in the observed results.

6 Conclusion

This study examined the impact of instruction tuning on XLM-R models in the context of prompted classification across multiple SuperGLUE tasks. Our findings confirm that instruction tuning provides substantial benefits, particularly for models with more than $500M$ parameters. However, the performance gains do not directly scale with model size, challenging the assumption that larger models inherently benefit more from instruction tuning. Our observations suggest that pre-training setup may play a crucial role in determining the effectiveness of instruction tuning. While XLM-R_{XXL} achieves the strongest improvements, XLM-R_{large} remains competitive despite being significantly smaller. In contrast, XLM-R_{XL} underperforms, despite having nearly seven times the parameters of the large model. This suggests that pre-training compute and dataset diversity play a critical role in how well a model adapts to instruction tuning. Insufficient pre-training appears to limit the ability of larger models to fully leverage instruction-tuned data, reinforcing findings from broader LLM research on the importance of high-quality, diverse pre-training corpora.

Although our study offers meaningful insights into instruction tuning for encoder-based models, several aspects merit further investigation. Future work should investigate the relationship between pre-training characteristics and instruction tuning effectiveness, especially for larger models. A more systematic analysis of these factors could help disentangle their specific contributions. Additionally, our instruction tuning dataset, though effective, is limited in diversity. Expanding the range of instruction tuning tasks may improve generalization and provide a broader perspective on how data diversity shapes model adaptation. Addressing these questions will further refine our understanding of instruction tuning and its role in enhancing encoder-based models.

Limitations

While this study provides valuable insights into the impact of instruction tuning on XLM-R models for prompted classification, there are several important limitations to consider. One key limitation is the dependency of instruction tuning effectiveness on the quantity and quality of pre-training data. Despite experimenting with large models like XLM-R_{XL} and XLM-R_{XXL}, we observed limited improvements, likely due to insufficient pre-training data. This suggests that the benefits of instruction tuning may be contingent on having sufficiently large and diverse datasets, which was not fully explored in this study.

Additionally, the use of heavily filtered instruction data may have constrained the models' ability to generalize effectively. The narrow range of instructions used could have limited the adaptability of the models to more complex or diverse tasks. Expanding the diversity of instruction data could help

address this limitation and improve generalization. Our focus on SuperGLUE classification tasks also limits the generalizability of our findings, as these tasks may not fully capture the broader capabilities of instruction-tuned models across other task types, such as question answering or text generation. Furthermore, we did not extensively explore the zero-shot learning potential of instruction-tuned models. Finally, our study was limited to the XLM-R family of models, and the findings may not generalize to other encoder architectures.

Acknowledgments

References

Benjamin Clavié, Nathan Cooper, and Benjamin Warner. 2025. It's all in the [mask]: Simple instruction-tuning enables bert-like masked language models as generative classifiers. *Preprint*, arXiv:2502.03793.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *Preprint*, arXiv:1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Gemini Team et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-scale transformers for multilingual masked language modeling. *Preprint*, arXiv:2105.00572.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need. *Preprint*, arXiv:2306.11644.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Pranjal Kumar. 2024. Large language models (llms): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, 57(10):260.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2022. A survey of transformers. *AI Open*, 3:111–132.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *Preprint*, arXiv:2006.04884.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. *Preprint*, arXiv:2211.01786.

Josh Achiam et al. OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Teven Le Scao and Alexander M. Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2627–2636. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

Trieu H. Trinh and Quoc V. Le. 2019. A simple method for commonsense reasoning. *Preprint*, arXiv:1806.02847.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. Superglue: A stickier benchmark for general-purpose language understanding systems. *Preprint*, arXiv:1905.00537.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024a. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *Preprint*, arXiv:2412.13663.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024b. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *Preprint*, arXiv:2412.13663.

Chuhan Wu and Ruiming Tang. 2024. Towards a universal scaling law of llm training and inference. *ScienceOpen Preprints*.

Yisheng Xiao, Juntao Li, Zechen Sun, Zechang Li, Qingrong Xia, Xinyu Duan, Zhefeng Wang, and Min Zhang. 2024. Are bert family good instruction followers? A study on their potential and limitations. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

Qiang Zhang, Keyan Ding, Tianwen Lv, Xinda Wang, Qingyu Yin, Yiwen Zhang, Jing Yu, Yuhao Wang, Xiaotong Li, Zhuoyi Xiang, et al. 2024. Scientific large language models: A survey on biological & chemical domains. *ACM Computing Surveys*.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. 2021. Revisiting few-sample bert fine-tuning. *Preprint*, arXiv:2006.05987.

Yang Zhao, Li Du, Xiao Ding, Kai Xiong, Zhouhao Sun, Jun Shi, Ting Liu, and Bing Qin. 2024. Deciphering the impact of pretraining data on large language models through machine unlearning. *Preprint*, arXiv:2402.11537.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. 2023. Pytorch fsdp: Experiences on scaling fully sharded data parallel. *Preprint*, arXiv:2304.11277.

A Average Prompt Advantage

To evaluate the effectiveness of instruction tuning, we initially considered the *average prompt advantage* metric introduced by Scao and Rush (2021). However, upon closer analysis, we identified significant discrepancies between the scores and the trends observed in our plots. Consequently, we opt not to rely on this metric.

Table 3 presents the average prompt advantage scores across all experiments. An example of the metric's limitations can be observed in the *WiC* dataset. The highest reported score is for XLM-R_{large}, with a value of 78.3. However, visual inspection of our plots clearly indicates that instruction tuning provides the most substantial improvement for the XLM-R_{XXL} model, which only achieves a score of 39.2 according to the metric.

This discrepancy arises due to the intrinsic properties of the *average prompt advantage* metric. Specifically, the metric only considers the y-band where both the instruction-tuned and prompt-based models have defined values. This restriction leads to an underestimation of improvements when the defined range is narrow. In the case of XLM-R_{XXL}, the improvement due to instruction tuning extends beyond this constrained band, resulting in a severe underestimation of the true advantage gained.

Given these limitations, we conclude that the *average prompt advantage* metric does not adequately reflect the benefits of instruction tuning, particularly for models where the overall improvement is substantial but distributed beyond the defined evaluation band. As a result, we refrain from using this metric for reporting our findings.

B Estimating Pretraining Compute

To estimate the compute invested in pretraining for the different XLM-R models, we followed an empirical approach based on hyperparameters reported by Goyal et al. (2021). Specifically, we used the reported batch size and sequence length for each model.

We simulated pretraining by generating a random dataset and running a single pretraining step with these hyperparameters. During this step, we tracked the number of floating-point operations (FLOPs) required. Using the number of training tokens processed in this single step, we then extrapolated the total FLOPs for full pretraining based on the total number of training tokens reported by Goyal et al. (2021). Table 4 provides insights into

Model	BoolQ	CB	COPA	RTE	WiC	Average
XLM-R_{base}	24.4 ± 23.8	-12.4 ± 30.3	16.3 ± 62.6	17.2 ± 16.6	3.7 ± 44.1	9.9 ± 35.6
XLM-R_{large}	38.6 ± 56.6	$\mathbf{1.1 \pm 22.0}$	27.4 ± 24.7	$\mathbf{58.4 \pm 23.3}$	$\mathbf{78.3 \pm 46.0}$	$\mathbf{40.8 \pm 41.7}$
XLM-R_{xl}	$\mathbf{46.6 \pm 21.6}$	-1.0 ± 12.2	27.1 ± 29.9	45.6 ± 39.6	44.4 ± 38.5	32.5 ± 31.8
XLM-R_{xxl}	41.0 ± 38.1	-0.2 ± 17.6	$\mathbf{39.7 \pm 24.5}$	43.9 ± 18.3	39.6 ± 40.3	32.8 ± 30.2

Table 3: Average prompt advantage of the instruction-tuned XLM-R models over their prompt-based base models.

Model	Batch Size	Sequence Length	1-Step #Tokens	1-Step #FLOPs
XLM-R_{large}	8192	512	4.19×10^6	1.53×10^{16}
XLM-R_{xl}	2048	512	1.05×10^6	1.79×10^{16}
XLM-R_{xxl}	2048	512	1.05×10^6	6.10×10^{16}

Table 4: Compute measurements for a single pretraining step using hyperparameters from Goyal et al. (2021).

the compute measurements for one step. By extrapolating these values based on the total number of training tokens, we obtained the estimated FLOPs reported in Table 1.

C Additional Result Plots

Figure 5 shows the performance curves for the *BoolQ* and *RTE* tasks.

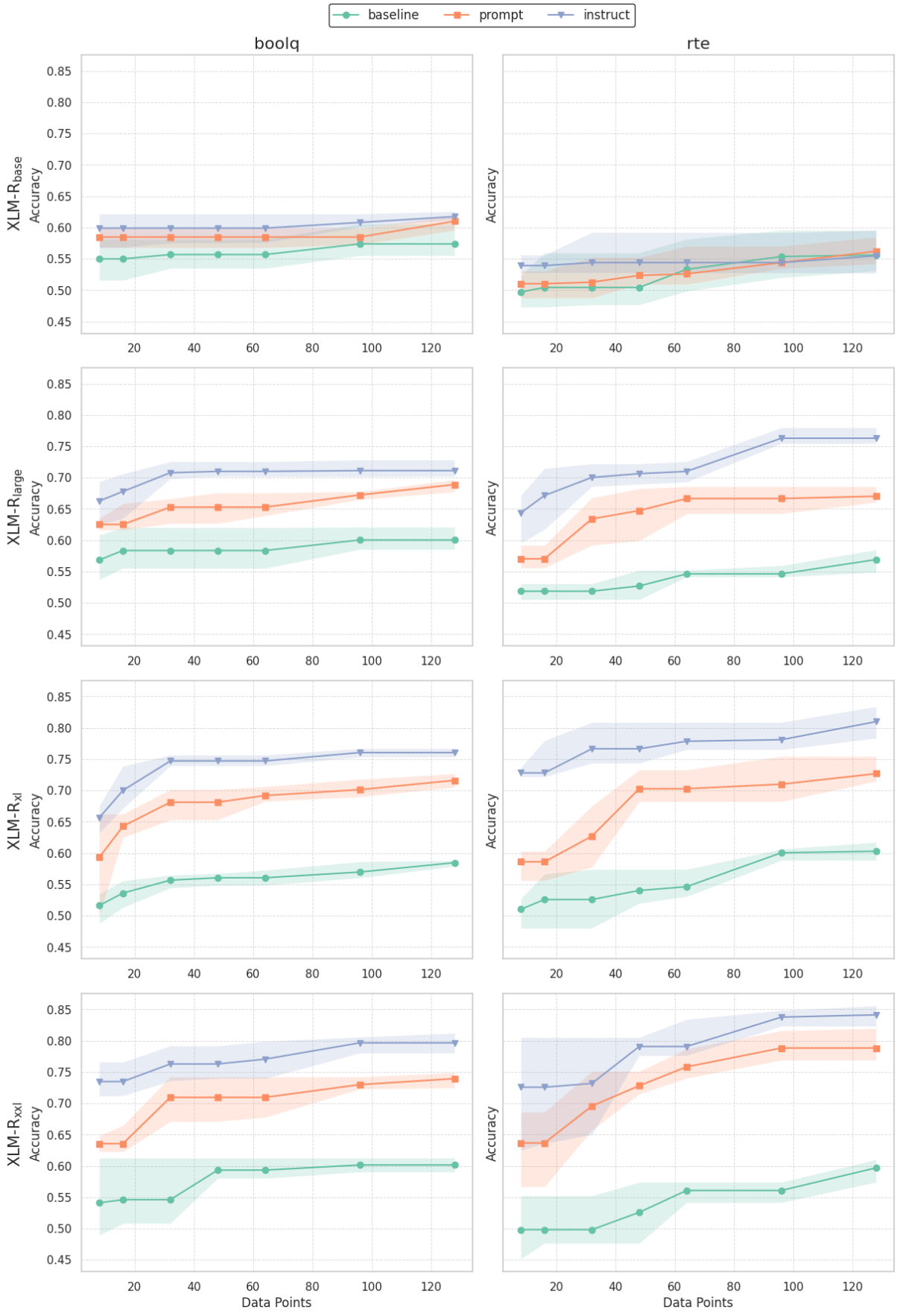

Figure 5: Performance curves of XLM-R model sizes on the *BoolQ* and *RTE* benchmarks, comparing different fine-tuning strategies.