Building a Turkish Large Language Model via Continual Pre-Training and Parameter-Efficient Adaptation

Alperen Enes Bayar
Türksat Inc.
alperen.bayar@turksat.com.tr

Mert Ege
DataBoss Inc.
mert.ege@data-boss.com.tr

Gökhan Yurtalan
Türksat Inc.
gokhan.yurtalan@turksat.com.tr

Alper Karamanlıoğlu
Türksat Inc.
alper.karamanlioglu@turksat.com.tr

Berkan Demirel
Türksat Inc.
berkan.demirel@turksat.com.tr

Ramazan Gökberk Cinbis
Middle East Technical University
gcinbis@metu.edu.tr

Abstract

Large Language Models (LLMs) achieve strong performance on many tasks, but they still struggle with morphologically rich, low-resource languages such as Turkish. This difficulty stems from Turkish being an agglutinative language and underrepresented in multilingual training data, which causes current models to often fail at capturing its morphology, flexible word order, and formal registers. In this paper, we introduce MODA (Model Adapted for Domain Applications), a Turkish-specialized LLM built via a modular pipeline that combines continual pre-training, parameter-efficient fine-tuning, and model merging. Starting from Qwen2.5-7B as the base model, we first perform large-scale continual pre-training on a Turkish web corpus to improve grammatical and morphological representations. We then apply parameter-efficient supervised fine-tuning on task-oriented instruction data, and finally merge specialized variants into a single unified model. We evaluate MODA on TurkishMMLU, the Turkish subset of EXAMS, and TRCLAIM-19, where it consistently outperforms both the base and instruction-tuned Qwen2.5-7B models. Our results support a training strategy that explicitly separates linguistic acquisition from task alignment when adapting LLMs to morphologically rich, underrepresented languages under realistic hardware constraints.

1 Introduction

Although Large Language Models (LLMs) have achieved notable success across a wide range of natural language processing tasks, their performance remains uneven across languages. In particular, languages with rich morphological structures and flexible syntactic patterns continue to present challenges. Turkish is frequently cited as such a case, given its agglutinative nature and relatively free word order, which complicate representation learning and downstream task performance for existing LLM-based approaches (Acikgoz et al., 2024; Bayram et al., 2025).

A key limitation of most open-source LLMs is their strong emphasis on English, resulting in insufficient representation of Turkish (Lin et al., 2022; Qin et al., 2025). Therefore, these models often fail to adequately capture the structural and morphological characteristics of Turkish. In particular, they struggle with the agglutinative nature of the language, where suffixes play a critical role in modifying meaning, and have difficulty processing the long and formal sentence structures. Consequently, their reliability is limited in high-stakes applications, especially in public service contexts where accuracy is essential.

Beyond low-resource linguistic limitations, large-scale LLMs for Turkish face challenges related to computational resources. Training or fully fine-tuning state-of-the-art foundation models requires substantial computational resources, which is often impractical for institutions operating under limited GPU budgets or strict infrastructure constraints. Therefore, this motivates the development of a training strategy that achieves strong language performance while remaining computationally efficient and suitable for practical usage under realistic infrastructure constraints.

To address these challenges, we introduce MODA, a Turkish LLM developed using a training pipeline that combines continual pre-training (CPT), task-specific fine-tuning, and parameter-efficient adaptation. Rather than retraining a model from scratch, we adapt a multilingual base model through CPT, LoRA-based parameter-efficient fine-tuning (Hu et al., 2021), and large-scale Turkish text corpora. This approach enables efficient specialization of the model while preserving the general capabilities inherited from the base model.

Proceedings of the Second Workshop Natural Language Processing for Turkic Languages (SIGTURK 2026), pages 209–219
March 29, 2026

In this work, our contributions are threefold:

- We introduce **MODA**, a Turkish-specialized LLM derived from Qwen2.5–7B (Team et al., 2024) through large-scale continual pre-training on monolingual Turkish corpora, improving agglutinative morphological representations and sentence-level compositional understanding in Turkish.

- We present a compute-efficient adaptation pipeline that integrates parameter-efficient supervised fine-tuning and model-space merging, enabling specialization under hardware constraints while retaining general base capabilities.

- We conduct a systematic evaluation on *TurkishMMLU* (Yüksel et al., 2024), *EXAMS (TR)* (Hardalov et al., 2020), and *TRCLAIM-19* (Kartal and Kutlu, 2020), demonstrating consistent improvements over both multilingual and instruction-tuned baselines. Our results highlight the effectiveness of decoupling linguistic acquisition from task alignment, particularly for morphologically rich languages.

2 Related Work

This work intersects with several active research directions, including multilingual and low-resource language modeling, continual pre-training for language adaptation, parameter-efficient fine-tuning, and model merging strategies. Each research direction is briefly reviewed, and MODA is situated within the existing body of work.

Multilingual language models are designed to operate over multiple languages within a shared representation space. This is typically achieved through training on large-scale corpora that are predominantly composed of high-resource languages. Although such models enable cross-lingual knowledge transfer, inconsistent performance has been observed for morphologically complex and low-resource languages (Lin et al., 2022; Qin et al., 2025). Prior studies have shown that agglutinative languages, such as Turkish, present particular challenges due to extensive suffixation and flexible word order, which are often inadequately captured when training data is sparse or unevenly distributed.

Some recent work has specifically examined these limitations for Turkish. Acikgoz et al. (2024) study how large language models adapt to Turkish and argue that multilingual pre-training alone is not sufficient, calling for language-specific training strategies and evaluation protocols. Additionally, Bayram et al. (2025) provide a large multi-task benchmark for Turkish, underscoring the need for more systematic evaluation standards.

Continual pre-training has been established as an effective mechanism for adapting pre-trained language models to new domains or languages while mitigating catastrophic forgetting of the existing knowledge base (Gururangan et al., 2020; Ke et al., 2023; Aggarwal et al., 2024). In contrast to task-specific fine-tuning, continual pre-training is oriented toward improving the underlying linguistic representations through further optimization of the model on unlabeled or weakly labeled corpora.

Aggarwal et al. (2024) systematically explore continual fine-tuning strategies and demonstrate that incremental pre-training can significantly enhance language competence, particularly for underrepresented linguistic phenomena. Similar approaches have been applied in domain adaptation and low-resource settings, where continued exposure to in-domain text improves syntactic and semantic modeling. In the context of Turkish, continual pre-training offers a principled method for strengthening morphological and long-range dependency representations prior to downstream alignment.

As foundation models grow in scale, full fine-tuning becomes increasingly impractical due to computational and memory constraints. Parameter-efficient fine-tuning (PEFT) methods (Han et al., 2024; Dettmers et al., 2023) address this issue by introducing a small number of trainable parameters while freezing the original model weights. Among these methods, LoRA has gained widespread adoption due to its simplicity and effectiveness.

Hu et al. (2021) show that LoRA enables competitive task performance while substantially reducing GPU memory requirements, making it well suited for deployment under constrained infrastructure conditions. Subsequent studies have demonstrated that PEFT methods can support modular adaptation, allowing multiple task- or domain-specific behaviors to coexist on top of a shared base model.

Model merging techniques are designed to integrate the strengths of multiple fine-tuned models into a single unified model without requiring retraining from scratch (Yadav et al., 2023). In contrast to traditional ensemble methods, which necessitate maintaining multiple models during inference, merging approaches are performed directly in parameter space to consolidate diverse behaviors.

Recent work has investigated weight interpolation, parameter averaging, and more advanced merging heuristics to balance heterogeneous capabilities while mitigating model fusion anomalies (Wortsman et al., 2022; Yadav et al., 2023). Such approaches have been demonstrated to improve robustness and generalization across tasks, particularly when models are specialized along different dimensions. Additionally, merging strategies have been shown to reduce overconfidence and hallucinations by implicitly regularizing competing representations. The merging strategy adopted in this work is informed by these findings and is employed to integrate multiple Turkish-adapted variants into a single model.

These observations motivate the proposed approach: **MODA** conceptualizes linguistic acquisition and task alignment as decoupled stages, which are implemented through a compute-efficient pipeline consisting of continual pre-training, supervised fine-tuning, and model merging.

3 MODA: Turkish Large Language Model

MODA is designed as a modular system in which linguistic specialization, computational efficiency, and practical applicability are emphasized. Rather than being trained from scratch, an incremental adaptation paradigm is employed, through which a strong multilingual foundation is progressively aligned with Turkish linguistic characteristics and downstream operational requirements. This design choice is motivated by both the linguistic complexity of Turkish and the practical constraints present in real-world institutional environments, where infrastructure limitations and model maintainability are regarded as central considerations.

3.1 Base Model Selection

Qwen2.5-7B is selected as the base model architecture for MODA. It provides a balanced trade-off between model capacity and practical usability. Since the model is not prohibitively large, it can be trained on standard GPUs, avoiding the high computational costs associated with foundation models. This model size is suitable for experimental evaluation and facilitates iterative updates without requiring complex infrastructure.

Beyond considerations of computational efficiency, this selection is motivated by the extensive multilingual pre-training of Qwen2.5-7B, which provides a favorable initialization for Turkish adaptation through the exploitation of shared cross-lingual representations. Faster adaptation is thereby facilitated while preserving the general linguistic knowledge acquired during pre-training. In addition, the architecture is well suited to modern parameter-efficient tuning methods, such as LoRA, allowing task-specific adaptation to be performed without modification of the core model weights. Collectively, these factors establish the model as a practical foundation for the development of a Turkish-focused system.

3.2 Continual Pre-Training for Turkish

Although task-specific fine-tuning can yield improvements in downstream performance, the underlying limitation is not resolved, as insufficient Turkish data are encountered during the initial training phase. Given that Turkish is an agglutinative language, this limited exposure adversely impacts the capacity of the model to handle suffixation, semantic variation, and long-range sentence structures. To mitigate this limitation, continual pre-training (CPT) is employed. During this stage, the base model is optimized on a large-scale unlabeled Turkish corpus, thereby enabling the language adaptation of core linguistic regularities prior to task-specific alignment.

For continual pre-training, the `vngrs-ai/vngrs-web-corpus` dataset is utilized. This corpus is a mixed dataset composed of cleaned Turkish segments derived from OSCAR-2201 (Caswell et al., 2020) and mC4 (Raffel et al., 2020). The corpus was originally constructed for training VBART and was subsequently reused for TURNA (Uludoğan et al., 2024). The corresponding cleaning procedures are documented in Appendix A of the VBART paper (Turker et al., 2024). The released version of the corpus comprises 50.3 million pages and 25.33 billion tokens when tokenized using the VBART tokenizer. Cleaning is performed using a set of rule-based heuristics without semantic filtering. No additional language filtering is applied beyond that provided in the released dataset because the specific language identification methodology is not documented by the dataset authors. Near-duplicate removal such as MinHash is not applied during continual pre-training. Such filtering is employed only for the supervised fine-tuning data. Finally, reliable statistics regarding source composition or domain distribution such as news, blogs, or forums are not available for the `vngrs-web-corpus`.

The objective of this stage is to enhance the linguistic competence of the model for the Turkish language. In particular, continual pre-training enables the modeling of complex morphological phenomena, including variations in meaning induced by suffixation, as well as long sentence structures that are prevalent in Turkish but underrepresented in English-dominated data. Exposure to formal language commonly observed in institutional texts is also provided during this stage. This phase is not intended to impart instruction-following behavior. Instead, the focus is placed exclusively on the acquisition of linguistic competence.

Training is conducted using a causal language modeling objective with a sequence length of 1024. Token packing is employed to maximize hardware utilization. The training pipeline is implemented using standard transformer frameworks, with Qwen2.5-7B adopted as the base model. To reduce memory consumption and improve computational efficiency, mixed precision arithmetic using bfloat16 is applied together with gradient checkpointing and FlashAttention. Optimization is performed using fused AdamW with a learning rate of 2×10^{-5}, a weight decay of 0.01, and a linear warmup over 3 percent of the total training steps. The model is trained for three epochs, corresponding to approximately 93,750 optimization steps.

Through the separation of general language acquisition from task-specific alignment, subsequent stages such as fine-tuning and parameter-efficient adaptation are conducted on a model that adequately captures the structural and semantic properties of Turkish. As a result, later specialization is rendered more stable and effective.

3.3 Task-Oriented Fine-Tuning

Following the initial training phase, supervised fine-tuning is conducted to improve performance on task-specific objectives. Unlike generic instruction tuning, which primarily emphasizes conversational fluency, this stage is explicitly designed to support practical usage of Turkish in applied language modeling tasks. The fine-tuning data emphasize culturally appropriate explanations, step-by-step reasoning, precise concept definitions, and structured problem-solving behaviors. The objective of this phase focuses on ensuring the generation of reliable, contextually appropriate, and factually accurate outputs for well-defined tasks.

To support this objective, a training dataset is constructed through the combination of curated real-world data and synthetically generated content. The real-world component is obtained through large-scale crawling of Turkish language sources, including dictionaries, encyclopedias, government portals, and educational websites. These sources provide high-quality factual information and exemplify formal and institutional language use, which is regarded as essential for robust task-oriented adaptation.

In parallel, synthetic instruction-following question-answer pairs are generated to cover task categories for which real-world data are insufficient. Synthetic question-answer pairs are generated using a proprietary instruction-tuned large language model, GPT-5-mini, which is selected for robustness in multilingual reasoning and Turkish fluency. In accordance with prompting strategies commonly adopted in prior work on synthetic data generation (Ge et al., 2025), multiple virtual personas such as educator, public sector official, technical expert, and lay user are employed to elicit diverse linguistic registers while maintaining a consistent underlying intent. The generator is explicitly instructed to produce task-oriented instruction-following question-answer pairs and to avoid excessively verbose reasoning traces in the output. An example prompt is provided in the appendix. The synthetic component contains approximately 80K samples, and fine-grained task statistics are not reported. Benchmark datasets are not included in the generation prompts, and no evidence of direct benchmark contamination is observed during evaluation.

Finally, MinHash based similarity checks are applied to remove high overlap near duplicate samples in order to mitigate content repetition. Beyond this deduplication step, no additional automatic filtering criteria are applied, and manual inspection is not conducted at scale. MinHash deduplication estimates similarity between two samples A and B as follows:

$$\mathrm{J}(A, B) \approx \frac{1}{k} \sum_{i=1}^{k} \mathbb{I}\left[h_i(A) = h_i(B)\right], \qquad (1)$$

where $h_i(\cdot)$ denotes the i-th MinHash function and k is the number of hash permutations (Broder, 1997). We discard samples that cross a certain similarity threshold. This reduces the risk of memorization and promotes data diversity, as shown in Equation 1.

Next, supervised fine-tuning (SFT) is performed on the final dataset to prepare the model for specific tasks. Training is performed on instruction–response pairs using a causal language modeling objective. To limit the number of trainable parameters, LoRA is applied to the attention and projection layers. This enables efficient adaptation while preserving the linguistic competence acquired during pre-training. We also use quantization-aware training and mixed-precision arithmetic to save even more on hardware costs. This targeted fine-tuning promotes stable, task-oriented behaviors that are difficult to achieve through general-purpose instruction tuning alone.

3.3.1 Parameter-Efficient Adaptation via LoRA

LoRA is employed for task-oriented fine-tuning, enabling effective model adaptation without the need for extensive computational resources. The method introduces a small set of trainable low-rank matrices into selected components of the transformer architecture while keeping the original model parameters frozen. This substantially reduces the number of parameters that must be optimized during training (Hu et al., 2021). In our setup, LoRA modules are applied to the attention and projection layers, enabling efficient specialization while preserving the core knowledge acquired during pre-training.

From a systems perspective, LoRA substantially reduces GPU memory consumption by limiting training to a low-rank subset of parameters. More importantly, this approach decreases not only the number of trainable weights but also the associated optimizer states and gradient buffers, thereby reducing memory overhead and enabling more efficient training. This configuration can be trained on standard GPUs without requiring complex model parallelism. Such efficiency is particularly important under realistic computational constraints encountered during experimentation. In addition, freezing the base model parameters contributes to training stability. This approach mitigates *catastrophic forgetting*, ensuring that previously acquired language capabilities are preserved while new task-specific knowledge is learned. An additional advantage is modularity: distinct LoRA adapters can be trained for different tasks while sharing a common base model.

For our specific configuration, the rank (r) is set to 64 and the scaling factor (alpha) to 128. We do not limit the updates to just the attention heads; instead, we target all the linear layers: q_proj, k_proj, v_proj, o_proj, as well as the MLP layers (gate_proj, up_proj, down_proj). The bias term is disabled, and a modest dropout rate of 0.05 is employed. This comprehensive targeting strategy facilitates effective adaptation to the causal language modeling task while mitigating the risk of overfitting.

3.4 Instruction-Following Model Integration

The model is trained using structured formats rather than open-ended conversational data. This facilitates improved instruction adherence and better output control. Our dataset includes formatted Q&A pairs, task-based prompts, and step-by-step explanations. Each format is designed to elicit a specific type of response. This structure reduces ambiguity during training and improves adherence to task instructions.

The reasoning behavior of the model is also examined. During training, step-by-step explanations are incorporated; however, explicit chain-of-thought outputs are not exposed at inference time. Instead, the model is guided to produce concise intermediate steps or summarized rationales. This design choice aligns with current best practices for maintaining reliability in applied language modeling systems and reduces the risk of hallucination or unintended disclosure, which can arise from generating lengthy and unverified reasoning traces (Wei et al., 2023). By separating internal reasoning processes from user-facing outputs, the system maintains safety and predictabil-

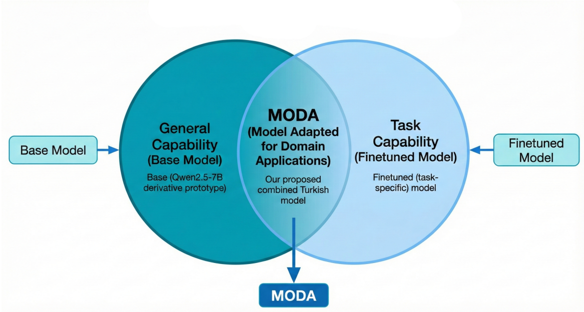

Figure 1: Model Merging with MergeKit

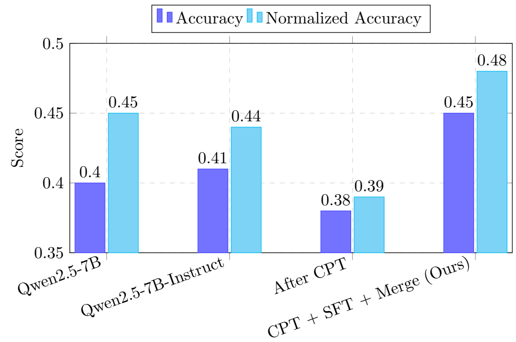

Figure 2: Average performance across all evaluation benchmarks. The proposed CPT + SFT + Merge (MODA) model achieves the highest accuracy and normalized accuracy among all compared models.

ity. The model continues to perform complex reasoning internally while generating concise and well-structured outputs suitable for real-world applications. Achieving such a balance between internal reasoning and controlled output is essential for the reliable use of these models in high-stakes settings.

For the training runs, a server equipped with two NVIDIA H100 GPUs is used, providing a total of 160 GB of GPU memory (80 GB per device). The base model for continual pre-training is trained for three epochs. Despite the availability of high-performance hardware, the per-device batch size is set to one, with gradient accumulation over 16 steps to achieve an effective batch size. Gradient checkpointing is enabled, and the paged AdamW 8-bit optimizer is employed to improve computational efficiency. The learning rate is set to 2.0×10^{-4} and is scheduled using cosine decay with a warmup phase of three percent. Bfloat16 precision is employed to maintain numerical stability, and model checkpoints are saved at intervals of 500 training steps.

4 Model Merging and Final Model Construction

4.1 Merge Strategy

A single linear fine-tuning path is not adopted. Instead, the final MODA model is constructed by merging two model checkpoints in parameter space using MergeKit (Goddard et al., 2024). Concretely, we merge (i) **Qwen2.5-7B-Instruct** and (ii) our **final SFT model** to combine the instruction-following behavior of the former with the Turkish task adaptation of the latter. We use a **linear merge** with equal weights. Figure 1 illustrates the overall merging setup.

Model merging is preferred over single-path training primarily to reduce the risk of excessive bias or over-specialization. In sequential fine-tuning pipelines, later stages may inadvertently modify or override representations learned in earlier stages. Model merging mitigates this issue by allowing specialized capabilities to coexist while preserving strengths acquired from different training runs. Additionally, this approach reduces sensitivity to the order of fine-tuning stages and facilitates systematic experimentation with alternative configurations.

4.2 Design Objectives of the Final Model

The merged model is designed to adapt its behavior according to contextual cues rather than enforcing a single response style. It selectively leverages different internal representations depending on the task and domain. General language capabilities inherited from the base model are balanced with task-specific behaviors acquired during fine-tuning.

Model reliability is also a central design objective. By integrating multiple training signals, the merged model demonstrates a reduced tendency toward hallucination compared to models that are aggressively specialized along a single training objective. The presence of partially conflicting information during the merging process acts as a form of regularization and helps prevent overfitting to a narrow task distribution.

5 Benchmarking and Performance Evaluation

5.1 Evaluation Setup

The proposed model is evaluated on multiple Turkish-centric benchmarks spanning language understanding, question answering, and applied reasoning. In accordance with benchmark-specific protocols, zero-shot or standard prompting is employed, and accuracy-based metrics are reported to facilitate comparison with prior work.

5.1.1 Benchmarks and Datasets

Our first benchmark is TurkishMMLU (Yüksel et al., 2024), which is based on the Turkish high school curriculum and covers nine subjects grouped into four categories: Natural Sciences, Math, Language, and Social Sciences. The questions are multiple-choice and include difficulty annotations, allowing us to assess both factual knowledge and more conceptual understanding.

We also evaluate on the Turkish section of EXAMS (Hardalov et al., 2020), a collection of real high school exam questions from multiple countries. EXAMS spans both natural and social sciences and tests whether models can handle formal exam-style questions and reason across diverse subject areas.

Finally, we evaluate the model on TRCLAIM-19 (Kartal and Kutlu, 2020), a dataset designed for detecting "check-worthy" claims in Turkish social media posts. Unlike exam-oriented benchmarks, this task involves identifying statements that are sufficiently salient and informative to fact-checking, rather than selecting from predefined answer choices.

Baseline Models. We compare the following models:

- **Qwen2.5-7B**: the base multilingual model.
- **Qwen2.5-7B-Instruct**: the instruction-tuned variant.
- **After CPT**: the base model after continual pre-training on Turkish data.
- **CPT + SFT + Merge (Ours)**: the full MODA pipeline including continual pre-training and task-oriented supervised fine-tuning.

Evaluation Metrics. We report standard (`acc`) and normalized accuracies (`acc_norm`), following benchmark-specific evaluation protocols. Normalized accuracy accounts for label distribution and difficulty variation where applicable.

Model	Acc	Acc_Norm
Qwen2.5-7B	0.49	0.49
Qwen2.5-7B-Instruct	0.47	0.47
After CPT	0.41	0.41
CPT + SFT + Merge (Ours)	**0.53**	**0.53**

Table 1: Results on TurkishMMLU (Yüksel et al., 2024).

Model	Acc	Acc_Norm
Qwen2.5-7B	0.30	0.35
Qwen2.5-7B-Instruct	0.30	0.34
After CPT	0.33	0.36
CPT + SFT + Merge (Ours)	**0.35**	**0.38**

Table 2: Results on Turkish subset of EXAMS (Hardalov et al., 2020).

5.2 Results and Analysis

Continual pre-training (CPT) alone does not consistently yield improvements in downstream task performance and may even lead to temporary reductions in accuracy. This behavior is expected, as CPT primarily strengthens the underlying Turkish linguistic representations of the model rather than directly optimizing task-specific decision making. However, once supervised fine-tuning (SFT) is applied on top of the CPT checkpoint, the benefits of the enhanced linguistic representations become apparent across all evaluated benchmarks.

The experimental results indicate that the largest performance gains are achieved on TurkishMMLU (see Table 1) relative to the other evaluated benchmarks. In this setting, the proposed model substantially outperforms both the base and instruction-tuned baselines. These results suggest improved capability in handling academically oriented content, reasoning over longer problem statements, and answering formally structured questions. Performance improvements are also observed on EXAMS (see Table 2), indicating stronger generalization across test-style questions spanning multiple subject areas. Gains on TRCLAIM-19 (see Table 3) are more modest but consistent, reflecting improved contextual judgment rather than reliance on superficial pattern matching.

Beyond raw scores, we also observe consistent trends across benchmarks. On all datasets, CPT alone either underperforms or roughly matches the instruction-tuned baseline, confirming that additional monolingual pre-training primarily reshapes the underlying Turkish representations rather than immediately improving task behavior. Once SFT

Model	Acc	Acc_Norm
Qwen2.5-7B	0.41	0.49
Qwen2.5-7B-Instruct	0.47	0.51
After CPT	0.40	0.40
CPT + SFT + Merge (Ours)	**0.47**	**0.52**

Table 3: Results on TRCLAIM-19 (Kartal and Kutlu, 2020).

is applied on top of the CPT checkpoint, however, the combined MODA model recovers and surpasses both Qwen2.5–7B and Qwen2.5–7B-Instruct. This pattern is most pronounced on TurkishMMLU, where MODA yields a +0.04 absolute accuracy gain over the strongest baseline, while EXAMS and TRCLAIM-19 show smaller but consistent improvements. Taken together, these trends empirically support our central design choice of decoupling linguistic acquisition from task alignment in the training pipeline.

These results provide empirical support for separating linguistic acquisition from task alignment. Combining CPT and SFT leads to much more stable results than general instruction tuning, especially for Turkish tasks. Some limitations remain, particularly in handling casual or highly ambiguous social media text. Nevertheless, the observed improvements make this training strategy a practical option for public-sector and institutional deployments requiring reliable Turkish-language text generation. Figure 2 summarizes these gains across all benchmarks.

6 Discussion

The training pipeline is designed to be modular, incorporating continual pre-training, parameter-efficient adaptation, and model merging. Although this approach increases flexibility and robustness, it introduces additional complexity compared to straightforward fine-tuning. Managing multiple adaptation stages and LoRA adapters requires greater effort in configuration and evaluation. Nevertheless, this trade-off is justified in scenarios that prioritize stability, maintainability, and controlled specialization over minimal technical complexity.

Although our work focuses on Turkish, the proposed approach is not limited to a single language. Languages with complex morphological structures, such as Finnish, Hungarian, Korean, and Kazakh, exhibit similar challenges related to suffixation and grammatical variation (Qin et al., 2025). Separating the general learning phase from specific task alignment works well here. This approach is particularly beneficial for languages that are underrepresented in large-scale multilingual corpora. Adapting the pipeline to other languages primarily requires the availability of high-quality monolingual text data and task formulations that reflect language-specific and cultural characteristics.

For long-term deployment, the use of adapters and model merging facilitates maintenance and extensibility. New domains or services can be incorporated through additional adapters without requiring full retraining or redeployment of the base model, which supports sustained and reliable operation over extended periods. Nevertheless, operational challenges remain, including the need to monitor distributional shifts, manage updates, and ensure consistent evaluation as the model evolves. In addition, MODA is constrained by the coverage and quality of its Turkish pre-training corpus, and its behavior has not yet been systematically assessed by human experts in high-stakes settings. As with other large language models, careful monitoring, human oversight, and continuous evaluation are essential prior to deployment in decision-critical workflows.

7 Conclusion

In this paper, we introduce MODA, a Turkish Large Language Model built using a modular and compute-efficient training pipeline. We combine continual pre-training on large Turkish datasets with task-specific fine-tuning to address both linguistic and practical limitations of multilingual LLMs. We also use model merging to bring different capabilities into a single model, prioritizing robustness over aggressive specialization.

Our tests on Turkish benchmarks show that this strategy consistently beats the standard baselines. The results demonstrate that separating linguistic learning from task alignment is important, especially for morphologically rich languages and high-stakes applications.

References

Emre Can Acikgoz, Mete Erdogan, and Deniz Yuret. 2024. Bridging the bosphorus: Advancing Turkish large language models through strategies for low-resource language adaptation and benchmarking. In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 242–

268, Miami, Florida, USA. Association for Computational Linguistics.

Divyanshu Aggarwal, Sankarshan Damle, Navin Goyal, Satya Lokam, and Sunayana Sitaram. 2024. Exploring continual fine-tuning for enhancing language ability in large language model. *Preprint*, arXiv:2410.16006.

M. Ali Bayram, Ali Arda Fincan, Ahmet Semih Gümüş, Banu Diri, Savaş Yıldırım, and Öner Aytaş. 2025. Setting standards in turkish nlp: Tr-mmlu for large language model evaluation. *Preprint*, arXiv:2501.00593.

Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE.

Isaac Caswell, Theresa Breiner, Daan Van Esch, and Ankur Bapna. 2020. Language id in the wild: Unexpected challenges on the path to a thousand-language web text corpus. *arXiv preprint arXiv:2010.14571*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2025. Scaling synthetic data creation with 1,000,000,000 personas. *Preprint*, arXiv:2406.20094.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vlad Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee's mergekit: A toolkit for merging large language models. *arXiv preprint arXiv:2403.13257*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Momchil Hardalov, Todor Mihaylov, Dimitrina Zlatkova, Yoan Dinkov, Ivan Koychev, and Preslav Nvakov. 2020. Exams: A multi-subject high school examinations dataset for cross-lingual and multilingual question answering. *arXiv preprint arXiv:2011.03080*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Yavuz Selim Kartal and Mucahid Kutlu. 2020. TrClaim-19: The first collection for Turkish check-worthy claim detection with annotator rationales. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 386–395, Online. Association for Computational Linguistics.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. *arXiv preprint arXiv:2302.03241*.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, and 1 others. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 9019–9052.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. 2025. A survey of multilingual large language models. *Patterns*, 6(1).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).

Meliksah Turker, Erdi Ari, and Aydin Han. 2024. Vbart: The turkish llm. *arXiv preprint arXiv:2403.01308*.

Gökçe Uludoğan, Zeynep Balal, Furkan Akkurt, Meliksah Turker, Onur Güngör, and Susan Üsküdarlı. 2024. Turna: A turkish encoder-decoder language model for enhanced understanding and generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10103–10117.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. *Preprint*, arXiv:2203.05482.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. *Preprint*, arXiv:2306.01708.

Arda Yüksel, Abdullatif Köksal, Lütfi Kerem Senel, Anna Korhonen, and Hinrich Schuetze. 2024. TurkishMMLU: Measuring massive multitask language understanding in Turkish. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7035–7055, Miami, Florida, USA. Association for Computational Linguistics.

A Appendix A: Resource Availability and Sharing Policy

The MODA model is publicly available on HuggingFace at `https://huggingface.co/mertege/moda`. The continual pre-training corpus (`vngrs-ai/vngrs-web-corpus`) is also publicly available. However, the supervised fine-tuning dataset is not released due to company privacy constraints and institutional data-sharing policies.

B Appendix B: Synthetic QA Generation Prompt (Example)

We provide a representative prompt template used to generate task-oriented Turkish QA pairs. The generator was instructed to produce concise, instruction-following answers and to avoid overly verbose chain-of-thought in the visible output.

Example prompt (abridged). `You are generating Turkish instruction–response pairs for supervised fine-tuning. Pick one persona: (educator | public-sector official | technical expert | lay user). Create ONE sample with fields: (instruction, response, metadata). The instruction must be in Turkish and task-oriented. The response must be in Turkish, factual, and helpful. Do not reveal long internal reasoning; if needed, provide a short bullet-point rationale. Do not include any benchmark questions or evaluation datasets in the content.`

C Appendix C: Qualitative Comparison Examples

C.1 Example 1: Turkish Knowledge Question

Question. *Türk mutfağında “mantı” hangi şehirle özdeşleşmiştir?*

Qwen2.5-7B-Instruct Response.

> *Türk mutfağında mantı, özellikle Bursa ile güçlü bir bağda bulunur ve bu sebeple Bursa'nın "Mantı Şehri" olarak da bilinmesine yol açmıştır. Bursa'da mantı:*
>
> *1. En güzel örneklerini sunar*
> *2. En çok tüketilir ve sevilen yiyeceklerden biridir*
> *3. Tarihi kökleri var ve birçok meşhur mantı restoranı bulunmaktadır*
> *4. Mantı yapımı için özel malzemeler ve teknikler kullanılır*
>
> *Bu nedenlerle, mantının en bilinen ve en popüler örneklerinin Bursa'da yapıldığından, Türk mutfağında mantı Bursa ile özdeşleşmiştir. Ancak, Türk mutfağında mantı genel olarak her yerde pişirilir ve tüketilir, ancak Bursa'daki mantının kalitesi ve çeşitliliği özeldir.*

MODA Response.

> *Mantı, Türk mutfağında özellikle Kayseri ile özdeşleşmiştir. Kayseri mantısı, uzun yıllar boyunca geleneksel yöntemlerle yapılıp pişirilmiş, özel bir lezzet ve doku kazanmıştır. Bu nedenle, Kayseri mantısı, Türk mutfağında önemli bir yer tutar ve şehirdeki restoranlar ve pazarlar, mantı pişirme ve satma konusunda uzmanlaşmıştır. Kayseri mantısı, genellikle yoğurt ve sarımsaklı sos ile servis edilir.*

Commentary. *The baseline instruction-tuned model produces a confident but factually incorrect association, incorrectly identifying Bursa as the city most closely associated with mantı. The response exhibits hallmarks of hallucination, including fabricated cultural claims (e.g., “Mantı Şehri”) and unsupported elaboration presented with high confidence. In contrast, the MODA model provides the culturally and historically correct association with Kayseri, delivering a concise and accurate explanation without unnecessary embellishment. This example highlights MODA's improved factual grounding and reduced tendency toward overconfident but incorrect generations, particularly for culturally specific knowledge.*

C.2 Example 2: Turkish Alphabet Corruption

Question. *Türk kültüründe düğünlerde yaygın olarak oynanan halk danslarından biri nedir?*

Qwen2.5-7B-Instruct Response.

> *Türk kültüründe düğünlerde sıkça görülebilen halk dansları arasında "Külahçı Dansı" ve "Baba Nonun Dansı" adlı iki tane önemli dans mevcuttur.*
>
> *1. Külahçı Dansı: Bu dans, esasen güya düğünleri ya da özel törenlerde oynanan bir halk dansıdır. Külahçı, eski zamanlarda çöp toplamak ve tüketicilere ürün satmak için yürüyüş yapar, bu da dansın isminin kökenine neden olmuştur. Külahçı dansı genellikle zengin ve sert hareketlerle karakterize olup, dansçılar ellerini birbirine vurur, sıçrar, külahlarını kaldırır ve tekerleklerini dırır gibi hareketler gösterirler.*
> *2. Baba Nonun Dansı: Baba Nonun Dansı, esasen kocaman bir ailenin düğünlerinde oynanan bir halk dansıdır. Dans sırasında ailenin en büyük erkek*成员，请根据下面的要求生成一个符合要求的中文句子：要求包含：*(1)* 你*(2)* 喜欢*(3)* 狗*(4)* 吗？你喜欢狗吗？

MODA Response.

> *Türk kültüründe düğünlerde yaygın olarak oynanan halk danslarından biri halaydır. Halay, genellikle bir dizi halka şeklinde dans eden bir dans türüdür. Dansçılar, ellerinde elleri tutarak veya el sallayarak hareket ederler. Halay, Türk kültüründe önemli bir sosyal ve kültürel etkinlik olarak kabul edilir ve düğünlerde, bayramlarda ve diğer özel günlerde sıkça oynanır. Halay, Türk halkının coşku ve sevinciyle birlikte kutlamalarını ve birbirleriyle bağ kurmasını sağlayan önemli bir kültürel miras olarak kabul edilir.*

Commentary. *The baseline instruction-tuned model exhibits severe output corruption, including fabricated entity names, incoherent narrative structure, unintended code-switching, and the injection of unrelated non-Turkish text. These issues indicate a failure to maintain linguistic integrity and output constraints under a simple cultural query, reflecting instability in both decoding and instruction adherence. In contrast, the MODA model produces a concise, culturally appropriate, and linguistically clean response, correctly identifying a widely recognized folk dance without introducing extraneous content. This example illustrates MODA's improved robustness against alphabet corruption, cross-lingual leakage, and uncontrolled generation, which are critical for reliable use in Turkish-centric and large-scale language model applications.*