Markhor: A framework for Adapting LLMs to Urdu via Alignment-Free Tokenizer Replacement

Anonymous ACL submission

Abstract

Large language models (LLMs) have established state-of-the-art performance across many NLP tasks, yet their effectiveness for low-resource languages such as Urdu remains limited due to under-representation in pretraining corpora and inadequate tokenization. This study explores adapting pretrained LLMs to new languages by replacing the original tokenizer without cross-lingual correspondence. We show that stable adaptation depends on maintaining architectural compatibility, matching vocabulary size, and reusing pretrained embedding and language modeling head weights. We propose *Markhor*, a framework for adapting LLMs to low-resource languages in this way. It spans tokenizer training, continual pretraining, and instruction tuning via knowledge distillation, resulting in two models, *MKGPT* and *MKQwen*, based on GPT and Qwen architectures. These models are evaluated on multiple classification tasks (fake news detection, hate speech detection, emotion classification) and open-ended question answering. Results show both MKGPT and MKQwen outperform baselines across all tasks, suggesting the effectiveness of the methodology for developing open-source Urdu LLMs. We release our datasets, models, and code at URL withheld.

1 Introduction

The performance of LLMs varies significantly across languages, with most open-source multilingual models giving subpar results for low-resource languages such as Urdu (Adeeba et al., 2025). While open-source LLMs provide greater accessibility and control, they often lag behind proprietary models in coverage, robustness, and reasoning ability, particularly in low-resource settings. For Urdu, limited representation in pretraining corpora results in poor vocabulary coverage and inefficient tokenization, as highlighted in recent benchmark studies (Arif et al., 2024; Tahir et al., 2024).

Adapting pretrained LLMs to a specific language is feasible, but the original tokenizer is often poorly suited to the target language. When applied to morphologically rich or underrepresented languages, it produces high-fertility tokens and fragmented representations, leading to inefficient encoding and degraded performance (Ali et al., 2024; Choudhury et al., 2025). This raises an important question: *can pretrained LLMs be effectively adapted to new languages by replacing their tokenizers without discarding or reinitializing the learned model parameters?* Addressing this is particularly important for low-resource languages, where retraining models from scratch is often computationally infeasible.

Prior work on cross-lingual transfer and tokenization adaptation has explored strategies such as alignment and vocabulary mapping to bridge the gap between source and target languages (Li et al., 2024; Ponti et al., 2019). These approaches preserve or transfer knowledge through explicit correspondences or reconstruction mechanisms. We investigate *complete tokenizer replacement under full vocabulary mismatch, without explicit cross-lingual alignment*. Our objective is not to replace alignment-based methods, but to examine whether stable adaptation is possible when cross-lingual resources or reliable mappings are unavailable.

To study this, we train Urdu-specific tokenizers and perform continual pretraining of existing LLMs under different initialization settings. Our analysis shows that stable adaptation depends on proper compatibility between the tokenizer and model parameters. In particular, mismatched vocabulary sizes or randomly initialized embedding and language modeling (LM) head layers lead to unstable or slow convergence. In contrast, maintaining the original vocabulary size and reusing pretrained embedding and LM head weights substantially improves training dynamics, resulting in faster convergence and lower loss values.

Building on this, we develop *Markhor*, a frame-

work for adapting LLMs to low-resource languages through data collection, tokenizer training, continual pretraining, and instruction tuning. For instruction tuning, we leverage knowledge distillation from strong teacher models, including GPT-4o-mini and DeepSeek, to improve instruction-following and reasoning capabilities. Thus, we develop two Urdu-specific LLMs based on GPT and Qwen architectures: *MKGPT* and *MKQwen*.

We evaluate the resulting models on five publicly available classification datasets spanning multiple tasks, including fake news detection, hate speech detection, and emotion classification. In addition, we introduce a multi-domain Urdu question answering (QA) benchmark to assess factual consistency and domain understanding. Experimental results show that our models consistently outperform the baselines across all tasks.

Our contributions are summarized as follows:

- We investigate adapting pretrained LLMs under tokenizer replacement without cross-lingual alignment and identify key factors that influence stable adaptation.
- We develop an end-to-end framework for Urdu LLM adaptation and introduce two Urdu-specific models, MKGPT and MKQwen, which achieve strong performance across multiple classification and QA benchmarks.
- We release a curated Urdu pretraining corpus, an instruction-following dataset, and a multi-domain QA benchmark to support future research in low-resource language modeling.

2 Related Work

This section reviews prior work on multilingual and Urdu language models, continual pretraining, tokenization, and knowledge distillation for LLMs.

Urdu and multilingual LLMs. Recent studies show that general-purpose multilingual LLMs remain substantially weaker in Urdu than in high-resource languages. Prior work reports that task-specific models often outperform prompting-based or zero-shot LLMs across multiple Urdu NLP tasks (Arif et al., 2024; Tahir et al., 2024). Kazi et al. (2025) show that Urdu QA remains highly sensitive to question type and linguistic complexity, highlighting weaknesses in reasoning. Although continual pretraining and instruction tuning can improve Urdu performance (Fiaz et al., 2025), limited public availability restricts reproducibility and broader analysis. Broadly, multilingual models such as BLOOMZ (Workshop et al., 2022), mGPT (Shliazhko et al., 2024), and Aya (Üstün et al., 2024) expand language coverage, but Urdu remains underrepresented in their pretraining corpora, vocabularies, and evaluation benchmarks. These findings suggest that off-the-shelf multilingual LLMs are insufficient for Urdu, motivating targeted adaptation of pretrained models.

Continual pretraining. Continual pretraining adapts pretrained models to target domains or languages at far lower cost than training from scratch (Jin et al., 2022). Prior work on domain-adaptive pretraining (DAPT) shows that further pretraining on in-domain corpora can greatly improve downstream performance (Gururangan et al., 2020; Ali et al., 2025), with similar benefits reported in broader domain- and task-adaptive settings (Huang et al., 2021; Xie et al., 2024). Most approaches keep the original tokenizer and vocabulary fixed, implicitly assuming that the pretrained segmentation scheme is already suitable for the target language. For Urdu, this assumption is problematic, since poor segmentation can directly weaken representation learning and limit the gains of further pretraining (Ali et al., 2024). This makes tokenization a central bottleneck in language adaptation, not just an implementation detail.

Tokenization and cross-lingual transfer. Tokenization is very important for morphologically rich and underrepresented languages because it affects segmentation quality and representation efficiency. Most multilingual models rely on shared subword vocabularies learned using methods such as byte-pair encoding (BPE) or unigram language models (Sennrich et al., 2016; Kudo and Richardson, 2018). While this supports broad multilingual coverage, it can lead to suboptimal segmentation and high token fertility for low-resource languages, including Urdu (Remy et al., 2024; Choudhury et al., 2025). To address this, several studies have explored tokenizer adaptation and cross-lingual transfer through embedding alignment and vocabulary mapping (Li et al., 2024; Ponti et al., 2019). Methods such as WECHSEL (Minixhofer et al., 2022) initialize target-language embeddings by aligning them with source-language embeddings, while other approaches rely on transliteration or vo-

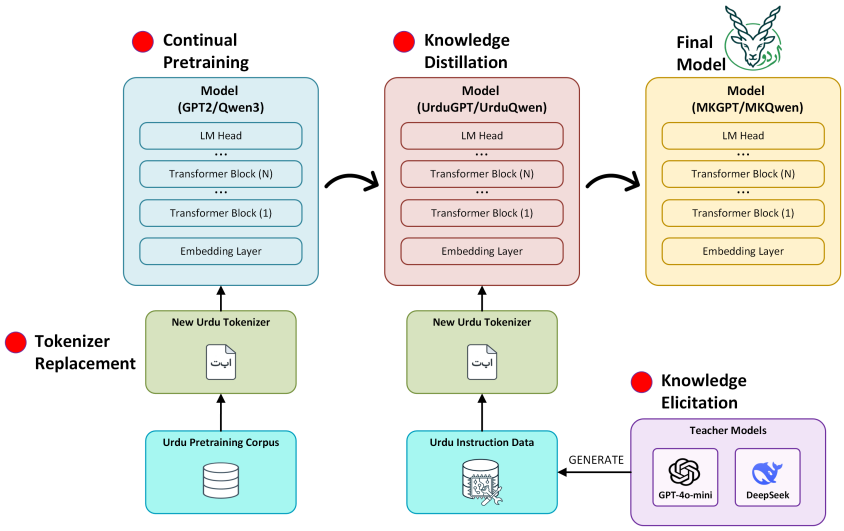

Figure 1: Four-stage Markhor framework for Urdu LLM training. **SFT**: Supervised Fine-Tuning. **BPE**: Byte-Pair Encoding.

cabulary mapping strategies (Artetxe et al., 2020). In contrast, we investigate tokenizer replacement under full vocabulary mismatch without explicit cross-lingual alignment. Rather than preserving token-level alignment, our approach relies on continual pretraining to relearn token representations while reusing pretrained model parameters, which may be helpful when cross-lingual resources or reliable mappings are unavailable.

Knowledge distillation for LLM adaptation. Knowledge distillation (KD) provides a practical way to compress and transfer knowledge from large models to smaller ones (Hinton et al., 2015). Early work such as DistilBERT (Sanh et al., 2019) and TinyBERT (Jiao et al., 2020) established effective KD frameworks for encoder models, whereas later work extended distillation to causal and instruction-tuned LLMs through teacher-student learning, self-distillation, and synthetic supervision (Gou et al., 2021; Yang et al., 2024; Shridhar et al., 2023; Yuan et al., 2024). In multilingual settings, KD has also been shown to transfer useful knowledge from high-resource to low-resource languages under limited target-language data (Gupta et al., 2023). In this work, KD serves as a downstream stage for knowledge enhancement and improved instruction following, complementing tokenizer adaptation and continual pretraining. Unlike teacher logits or divergence-based KD approaches (Xu et al., 2024), our framework performs black-box distillation through supervised fine-tuning (SFT) on teacher-generated Urdu instruction-response pairs.

3 Building Urdu LLMs

Figure 1 presents the four-stage framework for developing an Urdu LLM. Following data collection, we train an Urdu-specific tokenizer, then perform continual pretraining, knowledge elicitation, and distillation, resulting in the final Urdu LLMs. Each stage is detailed below.

3.1 Pretraining Data Collection

For continual pretraining, we curate a diverse Urdu corpus spanning encyclopedic, literary, and news domains, as summarized in Table 1. More details about data sources and preprocessing are provided in Appendix A.4. The final corpus contains ∼1.3 million Urdu articles and 1,368 books related to Urdu poetry and prose.

3.2 Tokenizer Training

We first trained an Urdu-specific tokenizer to address the high token-fertility (Ali et al., 2024) of existing multilingual tokenizers. On a held-out Urdu corpus, the original GPT-2 and Qwen3 tokenizers operate close to the character level, resulting in overly fragmented segmentation. Since both architectures employ BPE, we trained new BPE tokenizers on the collected Urdu corpus. Before training, we apply Unicode-based normalization to unify visually similar Urdu characters, remove optional diacritics and elongation (Tatweel) marks, and standardize punctuation and digits.

Design analysis. We conducted controlled experiments to analyze training dynamics under different

Data Source	Content Description	Examples	$\hat{T}$(UGPT)	$\hat{T}$(UQwen)
Urdu Wikipedia Crawl	Encyclopedic articles	135,000	32M	31M
CLE Urdu Books (Adeeba et al., 2014)	Prose and poetry books	1,368	49M	48M
Urdu News 1M (Hussain et al., 2021)	News articles	1,038,340	45M	43M
Jang & Dawn News (scraped)	News articles	100,000	18M	17M
BBC Urdu Science (scraped)	Science and factual reporting	960	1M	1M
Total			**145M**	**140M**

Table 1: Corpus used for pretraining. $\hat{T}$: $\sim$ token count, **UGPT:** UrduGPT, **UQwen:** UrduQwen.

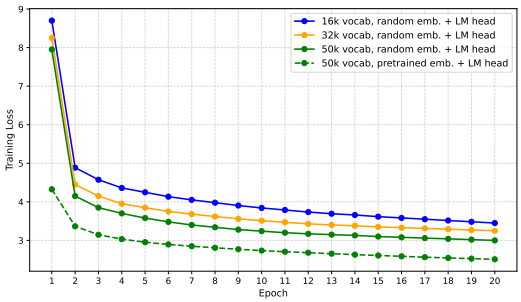

Figure 2: Training dynamics under tokenizer replacement (GPT-2). Colors denote vocabulary size (16k, 32k, and the original 50k), while line styles indicate initialization strategy (solid: random; dashed: pretrained embedding and LM head).

Tokenizer	Avg. Fertility (tokens/word)
GPT-2 (Original)	6.22
Qwen3 (Original)	5.43
GPT-2 (Urdu-trained)	1.59
Qwen3 (Urdu-trained)	1.16

Table 2: Avg. token fertility on the held-out Urdu corpus; higher values indicate near character-level segmentation (avg. word length $\sim$ 6 characters)

vocabulary sizes and initialization strategies using the setup described in Section 3.3. Specifically, we vary (i) the tokenizer vocabulary size with randomly initialized embedding and LM-head layers, and (ii) initialization of these layers (pretrained vs. random) while keeping the vocabulary size fixed.

Figure 2 shows that smaller vocabulary sizes lead to slower convergence and higher training loss, whereas matching the original vocabulary size results in more stable optimization. Also, reusing pretrained embedding and LM-head weights improves convergence over random initialization. Similar trends were observed for Qwen3; however, for brevity, we present only GPT-2 results. Accordingly, we keep the original vocabulary sizes (50,257 for GPT-2 and 151,643 for Qwen3), corresponding to roughly 49K and 150K merge operations, respectively, and adopt pretrained initializations for embedding and LM-head layers in all experiments. Table 2 shows substantially lower average tokens per word for the new tokenizers.

3.3 Continual Pretraining

In the second step, we tokenize our Urdu corpus using the new tokenizers and adapt the models under the causal language modeling (CLM) objective. Keeping the original vocabulary size allows reuse of pretrained weights, but the new tokenizer alters the token-to-ID mapping, requiring token representations to be relearned during continual pretraining. We therefore describe our approach as *continual pretraining with tokenizer adaptation*, where pretrained weights provide structural priors while token representations are relearned during training. This stage aims to improve Urdu next-token prediction and provide a stronger initialization for instruction tuning.

We employ two base architectures: GPT-2 and Qwen3. For GPT-2, three variants (117M, 345M, 774M) were evaluated, among which the 117M model showed the most stable convergence, likely due to a favorable balance between model capacity and data size (145M tokens). For Qwen3, the 0.6B variant was used due to computational constraints. GPT-2 has no prior exposure to Urdu, whereas Qwen3-0.6B includes limited Urdu coverage, verified through qualitative probing of zero-shot Urdu generation. Although replacing the tokenizer initially alters Qwen3's original Urdu capability, the downstream gains observed after continual pretraining suggest that the trade-off is beneficial for Urdu adaptation. The resulting models after this stage are *UrduGPT* and *UrduQwen*.

3.4 Knowledge Elicitation

The third stage involves knowledge elicitation using three teacher models (GPT-4o-mini, DeepSeek, and Llama3-70B) to generate Urdu instruction-

Domain	Subdomain	#Topics	DS	GPT
Science	Physics	78	1,193	8,418
	Chemistry	68	1,213	7,449
	Biology	63	1,204	6,858
	Earth Science	25	1,083	2,685
	Astronomy & Space Science	25	1,299	2,685
	Computer Science & Mathematics	25	1,165	2,671
	Technology	97	4,635	10,401
General	History	114	1,493	13,038
	Geography	76	1,207	8,320
	Famous Personalities	125	1,220	13,978
	Culture & Literature	20	984	2,216
	Sports	25	1,251	2,820
	Current Affairs	25	1,222	2,755
Total	—	**766**	**19,169**	**84,294**

Table 3: Distribution of instruction–response pairs generated across science and general knowledge domains by two teacher models: DeepSeek (DS) and GPT-4o-mini (GPT).

response pairs. Llama3-70B was later excluded due to quality issues. The objective was to obtain diverse, domain-rich samples spanning science and general knowledge. Given a domain, we prompted ChatGPT to generate 1,000 domain-related topics, which were expanded into subdomains to increase diversity and reduce duplication. After manual review and removal of redundant entries, 766 unique topics were retained, as shown in Table 3.

Each topic was dynamically incorporated into the prompt for instruction–response generation, with teacher models producing 50 Urdu pairs per topic. Due to token length constraints, some responses contained incomplete JSON objects, which were removed during post-processing.

Quality Assurance. Manual inspection showed that Llama3-70B often produced mixed Urdu-Chinese text, whereas GPT-4o-mini and DeepSeek generated higher-quality Urdu outputs; therefore, Llama3-70B was excluded from further use. Automatic filtering removed near-duplicate and low-content responses. Near-duplicates were identified using Jaccard similarity on token-level lexical overlap (threshold > 0.9), while low-content responses contained five words or fewer. In addition, outputs from GPT-4o-mini and DeepSeek were cross-compared during early generation stages to identify inconsistent or clearly incorrect samples. While large-scale factual verification was infeasible, these procedures ensured a linguistically robust and factually plausible instruction tuning dataset. The final dataset contains 103,463 instruction-response pairs (Table 3). GPT-4o-mini contributed a larger share of the data due to batch API support, whereas DeepSeek relied on single-request generation.

3.5 Knowledge Distillation

The final stage performs instruction tuning via SFT of UrduGPT and UrduQwen. Separate pipelines were used due to differences in architecture and chat formatting.

UrduGPT lacks a built-in chat template, so each instruction-response was flattened into plain text as: `Instruction:{instruction}\nResponse:{response}{eos_token}`. Instruction tuning was performed in two stages: DeepSeek data was used first, followed by GPT-4o-mini data to transfer more complex domain-specific knowledge. This staged training improved convergence stability.

UrduQwen Qwen3's chat template was applied to each instruction-response pair via the `apply_chat_template()` function. Messages followed a three-turn system-user-assistant structure compatible with the model's chat structure. Unlike UrduGPT, staged training was not applied because UrduQwen exhibited faster convergence during pretraining; therefore, instruction-tuning data from both teacher models was used jointly.

In both cases, models were optimized using next-token prediction loss over instruction-response pairs, distilling linguistic and task knowledge from the teacher models.

4 Experiments

4.1 Experimental Setup

Pretraining Setup. UrduGPT and UrduQwen were initialized from their respective base model weights and trained using mixed-precision (FP16/FP32). All parameters, including embedding, multi-head attention, and LM-head layers, were optimized end-to-end using AdamW with a learning rate of 5e-4, linear decay scheduling, and gradient accumulation over four steps. A sequence length of 512 was selected through empirical testing, while warmup steps were set to 5% of total training steps. Both models were trained for 20 epochs on four NVIDIA RTX A6000 (49 GB) GPUs using the `Accelerate` library.

Using their respective tokenizers, the Urdu corpus was tokenized into approximately 145 million tokens for UrduGPT and 140 million for UrduQwen, with a 90/10 train-validation split. Due to the slightly smaller token count, UrduQwen had

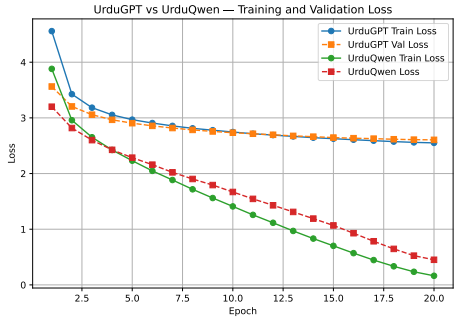

Figure 3: Pretraining learning curve of UrduGPT / UrduQwen.

fewer total and warmup training steps. Training checkpoints were maintained based on the lowest validation loss. Key hyperparameters, chosen through a series of empirical experiments, are listed in Table 6 (Appendix A.3).

Figure 3 compares the training and validation loss curves of UrduGPT and UrduQwen during continual pretraining. In both models, the validation loss remains lower than the training loss during early epochs, indicating effective regularization and early-stage underfitting. As training progresses, the training loss falls below the validation loss — around the 11th epoch for UrduGPT and much earlier for UrduQwen — suggesting closer fitting to the training data. Overall, both models exhibit a smooth decline in loss, indicating stable convergence. However, UrduQwen achieves substantially faster loss reduction and lower overall training and validation loss, likely due to its larger capacity and architectural improvements over UrduGPT.

Knowledge Distillation Setup. We applied SFT to UrduGPT and UrduQwen to distill instruction-following and domain knowledge from teacher models. UrduGPT was trained in two stages: first on DeepSeek data, then on GPT-4o-mini data. UrduQwen was trained in a single stage using the combined instruction corpus formatted with its built-in chat template. Both models were fine-tuned using the same hyperparameter configuration summarized in Table 7 (Appendix A.3). The only exception was UrduQwen's learning rate, reduced to 5e-6 to prevent rapid loss convergence and validation instability observed during early experiments. Each training session was run for 20 epochs using mixed-precision (FP16/FP32), retaining the checkpoint with the lowest validation loss. We refer to the final distilled models as *MKGPT* and *MKQwen*.

Figure 4 presents the instruction-tuning learning

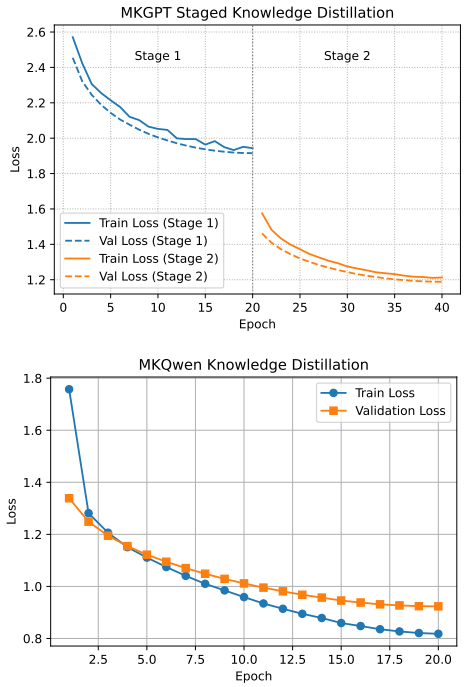

Figure 4: Instruction-tuning learning curves of MKGPT (top) and MKQwen (bottom).

curves for both models. For MKGPT, both training stages exhibit a sharp initial drop followed by gradual stabilization, indicating stable convergence and well-configured hyperparameters. The validation loss remains slightly below the training loss, suggesting effective generalization, with the final validation loss falling below 1.2. MKQwen also shows smooth convergence, although the validation loss exceeds the training loss in later epochs, indicating mild overfitting. However, the validation curve closely follows the training curve, with final losses around 0.8–0.9, reflecting stable learning.

4.2 Baselines

The original GPT-2 and Qwen3 models serve as primary baselines to assess the effectiveness of the proposed Markhor framework, including tokenizer replacement, continual pretraining, and knowledge distillation. We also evaluate multilingual encoder-based models, XLM-R and mBERT, on classification tasks (hate speech detection, fake news detection, and emotion classification), as they are widely used in prior Urdu NLP studies. For open-ended QA, we compare only against the original Qwen3 model. GPT-2 is excluded as it lacks instruction-following capabilities required for open-ended generation, while encoder-based models are not directly applicable to generative evaluation.

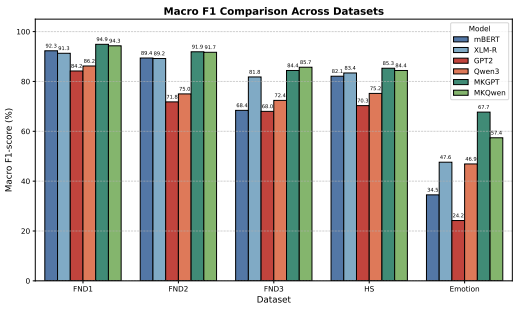

Figure 5: Macro-F1 (%, averaged over 5 runs) across five classification datasets. **FND**: Fake News Dataset, **HS**: Hate Speech.

Existing Urdu-specific instruction-tuned models such as UrduLlama (Fiaz et al., 2025) and Alif Instruct (Shafique et al., 2025) were initially considered. However, UrduLlama is not publicly available, while Alif Instruct suffered from missing dependencies and incomplete resources; therefore, both were excluded from comparison.

4.3 Evaluation

Datasets. For downstream evaluation, we use five publicly available Urdu datasets: one for hate speech detection (Akram et al., 2023), one for emotion classification (Bashir et al., 2023), and three for fake news detection (Harris et al., 2023; Amjad et al., 2020; Akhter et al., 2021) (Appx.A.1).

We further curate a supplementary QA dataset spanning five domains: physics, chemistry, biology, history, and geography. A native speaker created 100 QA pairs per domain with responses of at least 2-3 sentences, allowing topic diversity within each domain. Evaluation was performed on both *(i)* in-distribution examples — a held-out split of 100 QA pairs per domain from the instruction-tuning data — to assess knowledge retention, and *(ii)* out-of-distribution examples — a human-annotated dataset of equal size — to assess generalization and factuality.

Metrics. For classification tasks, we report precision, recall, F1-score, and accuracy, averaged across multiple runs as mean $\pm$ standard deviation. For generative evaluation, we report BERTScore-F1 to measure semantic similarity with gold responses. In addition, GPT-4o-mini serves as a judge to evaluate coherence, instruction adherence, vocabulary richness, and conceptual accuracy, assigning an overall score out of 100 for each response. To reduce potential bias, the evaluation prompt includes the question, reference answer, and candidate response (without specifying the model's name) to enable grounded and consistent comparisons across models.

Downstream Task Adaptation. For classification tasks, all models (MKGPT, MKQwen, GPT-2, Qwen3, XLM-R, mBERT) were fine-tuned using Hugging Face's `AutoModelForSequenceClassification`. Each dataset was split into 68% training, 12% validation, and 20% test sets. Maximum sequence lengths were selected based on dataset-specific token distributions. Models were trained for five epochs with learning rates between 1e-5 and 1e-6, selecting the configuration with the lowest validation loss. The `load_best_model_at_end` option was enabled to select the optimal checkpoint.

For open-ended QA, responses were generated using `model.generate()` with nucleus sampling ($p = 0.9$), top-k sampling ($k = 50$), temperature 0.8, repetition penalty 1.2, and a maximum of 200 new tokens. These settings aim to balance response coherence and diversity while reducing repetition.

4.4 Results and Discussion

Classification Tasks. Figure 5 presents the mean macro-F1 (averaged over 5 runs) of all models across five classification datasets.

MKGPT and MKQwen consistently outperform both encoder-based baselines (mBERT, XLM-R) and their respective base models (GPT-2, Qwen3), demonstrating the effectiveness of the proposed framework. Among the baselines, XLM-R generally performs better than mBERT, consistent with its stronger multilingual pretraining, while Qwen3 outperforms GPT-2, likely due to prior Urdu exposure during pretraining. Between the proposed models, MKGPT achieves slightly better overall performance than MKQwen. This may be attributed to its smaller parameter size, which is better suited to the relatively limited size of the evaluation datasets (each containing only a few thousand examples), resulting in improved generalization. MKQwen may require larger datasets to fully utilize its capacity.

In fake news detection (FND1–FND3), models achieve higher performance on the short-text dataset (FND1) compared to the long-article datasets (FND2 and FND3), suggesting that shorter, information-dense inputs are easier to model. Performance is weakest on the Emotion dataset, likely due to its smaller size (around 1,000 examples) and

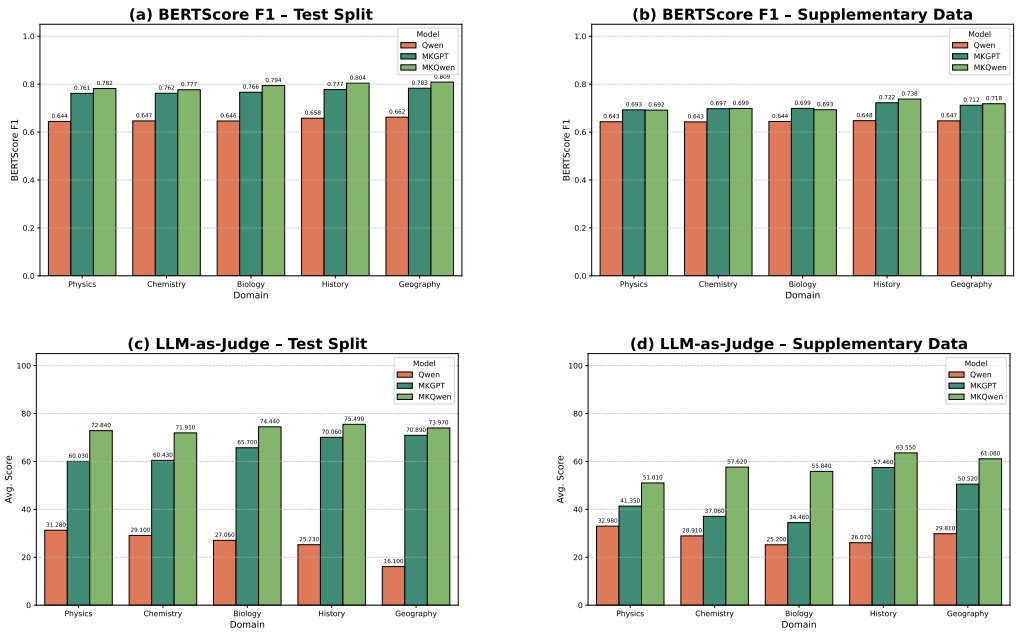

Figure 6: Question-answering evaluation results across domains. (a–b) BERTScore F1 on test and supplementary datasets. (c–d) LLM-as-Judge scores (aggregated out of 100) on test and supplementary datasets.

higher label complexity (six emotion categories).

Precision, recall, and accuracy follow similar trends. Full results, including mean ± standard deviation for all metrics, are given in the Appendix.

Question-Answering. Figure 6 compares the performance of MKGPT and MKQwen with the Qwen3 baseline on the QA task. Across both evaluation metrics, the proposed Urdu models consistently outperform the Qwen3 baseline. Across five domains, Qwen3 exhibits nearly constant BERTScore-F1 values ranging from 64.3% to 66.2%, while the mean LLM-as-judge scores vary between 16.1 and 32.98.

This is expected, as BERTScore measures semantic similarity between gold answers and generated responses, potentially awarding high scores to responses with correct wording but grammatically incorrect outputs, while LLM-as-judge evaluates coherence, instruction adherence, vocabulary richness, and conceptual correctness.

MKQwen generally outperforms MKGPT across all domains and splits, likely due to Qwen3's larger model size, prior Urdu exposure, and chat-oriented architecture, which provide a stronger foundation for continual pretraining and knowledge distillation. Domain-wise, both models perform better in history and geography than in science-related domains, which aligns with the nature of their continual pretraining corpus, which has limited science-related data. Science knowledge is primarily transferred to the models via distillation. Performance on supplementary human-annotated datasets is lower, possibly due to topic diversity beyond that seen during instruction tuning, highlighting the models' dependency on KD coverage.

5 Conclusion and Future Work

This study investigated whether pretrained LLMs can be adapted to new languages under complete tokenizer replacement without cross-lingual alignment. We introduced *Markhor*, a framework for adapting LLMs to low-resource languages, and used it to develop two Urdu-specific models, MKGPT and MKQwen. These models were evaluated on five classification datasets spanning three tasks and two QA test sets. Both consistently outperform their respective baselines, with MKGPT achieving stronger classification performance and MKQwen performing better on QA. Our findings demonstrate that effective language adaptation is possible even under full vocabulary mismatch, providing a practical alternative to alignment-based cross-lingual transfer. For future work, we plan to extend this approach to larger models, explore multilingual settings, investigate the role of human feedback and more diverse instruction data in further improving model capabilities, and conduct comparisons with alignment-based approaches.

Limitations

Data Coverage and Bias: Although care was taken during data collection and filtering, the training and evaluation data may still reflect topical and stylistic biases present in the available Urdu corpora and instruction-tuning sources. In particular, science-related content is relatively underrepresented during pretraining, which may limit performance in certain domains and affect generalization to unseen topics.

Model Size and Resource Constraints: Our models are comparatively small due to computational and data constraints, which restrict their capacity for complex reasoning and long-form generation. While this design choice enables efficient training for a low-resource language setting, it may limit performance on tasks that normally benefit from larger-scale models.

Scope of Evaluation Tasks: Evaluation is conducted on text classification and open-ended QA as these tasks are directly aligned with the supervision present in our Urdu instruction-tuning corpus. Tasks such as translation and abstractive summarization typically require a multilingual training corpus, while logical reasoning requires larger models trained with specialized objectives (Wei et al., 2022). Since our models are trained solely in Urdu and are comparatively small, these latter tasks fall outside the scope of the current work, though we plan to explore them in future work. In addition, large-scale human evaluation of generated responses was beyond the scope of the current study and remains an important direction for future work.

Ethical Statement and Broad Impact

Ethical Statement. The models developed in this work are trained using large-scale web and instruction data, which may contain societal, cultural, or topical biases. As a result, the generated outputs may reflect such biases, particularly in a low-resource language setting where curated resources are limited. Additionally, like other LLMs, our Urdu-specific models may produce fluent but factually incorrect responses. We therefore emphasize that these models are intended for research and assistive purposes, and that human oversight is essential in high-stakes or sensitive applications.

Broader Impact. This work contributes toward reducing the performance gap of LLMs for low-resource languages such as Urdu by releasing models, datasets, and training pipelines as open resources. Improved Urdu language technologies can support education, information access, and research for underrepresented communities. At the same time, increased accessibility to generative models raises concerns about misuse, including the generation of misleading or low-quality content. We encourage responsible use of the released resources and hope that this work will be a stepping stone for further research on robust, fair, and transparent language technologies for low-resource settings.

References

F. Adeeba, Q. Akram, H. Khalid, and S. Hussain. 2014. CLE Urdu Books N-Grams. In *Conference on Language and Technology*.

Farah Adeeba, Brian Dillon, Hassan Sajjad, and Rajesh Bhatt. 2025. Urblimp: A benchmark for evaluating the linguistic competence of large language models in urdu. *arXiv preprint arXiv:2508.01006*.

Muhammad Pervez Akhter, Jiangbin Zheng, Farkhanda Afzal, Hui Lin, Saleem Riaz, and Atif Mehmood. 2021. Supervised ensemble learning methods towards automatically filtering urdu fake news within social media. *PeerJ Computer Science*, 7:e425.

Muhammad Hammad Akram, Khurram Shahzad, and Maryam Bashir. 2023. Ise-hate: A benchmark corpus for inter-faith, sectarian, and ethnic hatred detection on social media in urdu. *Information Processing & Management*, 60(3):103270.

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Buschhoff, et al. 2024. Tokenizer choice for llm training: Negligible or crucial? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3907–3924.

Muhammad Zain Ali, Bernhard Pfahringer, and Tony Smith. 2025. Fake news classification in urdu: A domain adaptation approach for a low-resource language. *arXiv preprint arXiv:2512.22778*.

Maaz Amjad, Grigori Sidorov, Alisa Zhila, Helena Gómez-Adorno, Ilia Voronkov, and Alexander Gelbukh. 2020. "bend the truth": Benchmark dataset for fake news detection in urdu language and its evaluation. *Journal of Intelligent & Fuzzy Systems*, 39(2):2457–2469.

Samee Arif, Abdul Hameed Azeemi, Agha Ali Raza, and Awais Athar. 2024. Generalists vs. specialists:

Evaluating large language models for Urdu. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7263–7280, Miami, Florida, USA. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4623–4637.

Muhammad Farrukh Bashir, Abdul Rehman Javed, Muhammad Umair Arshad, Thippa Reddy Gadekallu, Waseem Shahzad, and Mirza Omer Beg. 2023. Context-aware emotion detection from low-resource urdu language using deep neural network. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(5):1–30.

Monojit Choudhury, Shivam Chauhan, Rocktim Jyoti Das, Dhruv Sahnan, Xudong Han, Haonan Li, Aaryamonvikram Singh, Alok Anil Jadhav, Utkarsh Agarwal, Mukund Choudhary, et al. 2025. Llama-3-nanda-10b-chat: An open generative large language model for hindi. *arXiv preprint arXiv:2504.06011*.

Layba Fiaz, Munief Hassan Tahir, Sana Shams, and Sarmad Hussain. 2025. Urdullama 1.0: Dataset curation, preprocessing, and evaluation in low-resource settings. *arXiv preprint arXiv:2502.16961*.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819.

Shivanshu Gupta, Yoshitomo Matsubara, Ankit Chadha, and Alessandro Moschitti. 2023. Cross-lingual knowledge distillation for answer sentence selection in low-resource languages. *arXiv preprint arXiv:2305.16302*.

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*.

Sheetal Harris, Jinshuo Liu, Hassan Jalil Hadi, and Yue Cao. 2023. Ax-to-grind urdu: Benchmark dataset for urdu fake news detection. In *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 2440–2447. IEEE.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Yinqiu Huang, Min Gao, Jia Wang, and Kai Shu. 2021. Dafd: Domain adaptation framework for fake news detection. In *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part I 28*, pages 305–316. Springer.

Khalid Hussain, Nimra Mughal, Irfan Ali, Saif Hassan, and Sher Muhammad Daudpota. 2021. Urdu news dataset 1m. *Mendeley Data*, 3.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 4163–4174.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2022. Lifelong pretraining: Continually adapting language models to emerging corpora. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4764–4780.

Samreen Kazi, Maria Rahim, and Shakeel Ahmed Khoja. 2025. Crossing language boundaries: Evaluation of large language models on urdu-english question answering. In *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 141–151.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 conference on empirical methods in natural language processing: System demonstrations*, pages 66–71.

Jiahuan Li, Shujian Huang, Aarron Ching, Xinyu Dai, and Jiajun Chen. 2024. Prealign: Boosting cross-lingual transfer by early establishment of multilingual alignment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10246–10257.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. Wechsel: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006.

Edoardo Maria Ponti, Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen. 2019. Cross-lingual semantic specialization via lexical relation induction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2206–2217.

François Remy, Pieter Delobelle, Hayastan Avetisyan, Alfiya Khabibullina, Miryam de Lhoneux, and Thomas Demeester. 2024. Trans-tokenization and cross-lingual vocabulary transfers: Language adaptation of llms for low-resource nlp. *arXiv preprint arXiv:2408.04303*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1715–1725.

Muhammad Ali Shafique, Kanwal Mehreen, Muhammad Arham, Maaz Amjad, Sabur Butt, and Hamza Farooq. 2025. Alif: Advancing urdu large language models via multilingual synthetic data distillation. In *Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025)*, pages 271–284.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Anastasia Kozlova, Vladislav Mikhailov, and Tatiana Shavrina. 2024. mgpt: Few-shot learners go multilingual. *Transactions of the Association for Computational Linguistics*, 12:58–79.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models. *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7059–7073.

Munief Hassan Tahir, Sana Shams, Layba Fiaz, Farah Adeeba, and Sarmad Hussain. 2024. Benchmarking the performance of pre-trained llms across urdu nlp tasks. *arXiv preprint arXiv:2405.15453*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. 2024. Efficient continual pre-training for building domain specific large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10184–10201.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.

Zhaorui Yang, Tianyu Pang, Haozhe Feng, Han Wang, Wei Chen, Minfeng Zhu, and Qian Liu. 2024. Self-distillation bridges distribution gap in language model fine-tuning. *arXiv preprint arXiv:2402.13669*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. 2024. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*.

Ahmet Üstün, Varun Aryabumi, Zhen Yong, Wei Yang Ko, Daniel D'souza, Gbolahan Onilude, Sara Hooker, et al. 2024. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939.

A Appendix

A.1 Downstream Datasets Statistics

Table 4 reports statistics of the datasets used for downstream classification. FND1–FND3 are sourced from Harris et al. (2023); Akhter et al. (2021); Amjad et al. (2020), while the hate speech and emotion datasets are taken from Akram et al. (2023) and Bashir et al. (2023), respectively. All datasets are reasonably balanced across classes.

Dataset	Content	#Labels	#Examples
FND1	Short headlines	2	10,083
FND2	Long articles	2	2,000
FND3	Long articles	2	1,300
HS	Tweets	2	21,760
Emotion	Tweets	6	1,121

Table 4: Statistics of datasets used for downstream classification tasks. **FND**: Fake News Dataset, **HS**: Hate Speech.

A.2 Detailed Classification Results

Table 5 shows detailed performance metrics for all models across the classification datasets used in our experiments, including accuracy and macro-averaged precision, recall, and F1-score, reported as mean $\pm$ standard deviation over 5 runs.

A.3 Detailed Hyperparameters

Tables 6 and 7 show the list of key hyperparameters used during pretraining and instruction tuning, respectively. These values were chosen after a series of empirical experiments performed to determine the optimal setting.

A.4 Pretraining Data Sources

Our primary sources included the publicly available Urdu Wikipedia crawl and the CLE Urdu Books corpus (Adeeba et al., 2014), which contains poetry and prose from well-known literary works. We further incorporated the Urdu News 1M dataset (Hussain et al., 2021), comprising one million Urdu news articles across four categories: business & economics, science & technology, entertainment, and sports. About 70% of these news articles were

Dataset	Model	Acc	Pre	Rec	F1
FND1	mBERT	0.928 ± 0.006	0.908 ± 0.009	0.939 ± 0.006	0.923 ± 0.006
	XLM-R	0.914 ± 0.007	0.901 ± 0.010	0.926 ± 0.008	0.913 ± 0.008
	GPT-2	0.857 ± 0.007	0.859 ± 0.009	0.844 ± 0.008	0.842 ± 0.007
	Qwen3	0.878 ± 0.004	0.881 ± 0.007	0.866 ± 0.006	0.862 ± 0.005
	MKGPT	$\mathbf{0.953 \pm 0.003}$	$\mathbf{0.951 \pm 0.004}$	$\mathbf{0.952 \pm 0.005}$	$\mathbf{0.949 \pm 0.003}$
	MKQwen	0.942 ± 0.004	0.945 ± 0.005	0.939 ± 0.005	0.943 ± 0.004
FND2	mBERT	0.892 ± 0.010	0.864 ± 0.013	0.926 ± 0.012	0.894 ± 0.011
	XLM-R	0.891 ± 0.012	0.852 ± 0.016	$\mathbf{0.938 \pm 0.011}$	0.892 ± 0.013
	GPT-2	0.719 ± 0.015	0.721 ± 0.016	0.717 ± 0.015	0.718 ± 0.015
	Qwen3	0.752 ± 0.013	0.761 ± 0.012	0.748 ± 0.013	0.750 ± 0.013
	MKGPT	$\mathbf{0.918 \pm 0.011}$	$\mathbf{0.922 \pm 0.010}$	0.914 ± 0.009	$\mathbf{0.919 \pm 0.008}$
	MKQwen	0.916 ± 0.009	0.919 ± 0.008	0.914 ± 0.008	0.917 ± 0.009
FND3	mBERT	0.623 ± 0.027	0.652 ± 0.049	0.727 ± 0.038	0.684 ± 0.030
	XLM-R	0.839 ± 0.015	0.829 ± 0.021	0.808 ± 0.019	0.818 ± 0.017
	GPT-2	0.707 ± 0.018	0.715 ± 0.017	0.679 ± 0.022	0.680 ± 0.021
	Qwen3	0.732 ± 0.015	0.725 ± 0.014	0.723 ± 0.015	0.724 ± 0.014
	MKGPT	0.849 ± 0.010	0.848 ± 0.010	0.841 ± 0.011	0.844 ± 0.010
	MKQwen	$\mathbf{0.859 \pm 0.009}$	$\mathbf{0.862 \pm 0.009}$	$\mathbf{0.855 \pm 0.010}$	$\mathbf{0.857 \pm 0.009}$
HS	mBERT	0.834 ± 0.011	0.835 ± 0.013	0.823 ± 0.012	0.821 ± 0.011
	XLM-R	0.842 ± 0.009	0.841 ± 0.010	0.835 ± 0.011	0.834 ± 0.008
	GPT-2	0.716 ± 0.016	0.704 ± 0.018	0.702 ± 0.019	0.703 ± 0.017
	Qwen3	0.768 ± 0.013	0.760 ± 0.014	0.747 ± 0.015	0.752 ± 0.014
	MKGPT	$\mathbf{0.861 \pm 0.009}$	$\mathbf{0.858 \pm 0.010}$	$\mathbf{0.850 \pm 0.011}$	$\mathbf{0.853 \pm 0.010}$
	MKQwen	0.852 ± 0.010	0.847 ± 0.011	0.841 ± 0.012	0.844 ± 0.011
Emotion	mBERT	0.378 ± 0.022	0.315 ± 0.034	0.423 ± 0.033	0.345 ± 0.033
	XLM-R	0.542 ± 0.021	0.574 ± 0.030	0.515 ± 0.028	0.476 ± 0.028
	GPT-2	0.291 ± 0.035	0.361 ± 0.038	0.267 ± 0.036	0.242 ± 0.034
	Qwen3	0.500 ± 0.028	0.538 ± 0.030	0.473 ± 0.029	0.469 ± 0.027
	MKGPT	$\mathbf{0.670 \pm 0.020}$	$\mathbf{0.683 \pm 0.021}$	$\mathbf{0.676 \pm 0.020}$	$\mathbf{0.677 \pm 0.020}$
	MKQwen	0.566 ± 0.025	0.633 ± 0.027	0.575 ± 0.026	0.574 ± 0.025

Table 5: Classification results reporting precision (Pre), recall (Rec), macro-F1, and accuracy (Acc), with mean $\pm$ standard deviation over 5 runs. **FND**: Fake News Dataset, **HS**: Hate Speech.

Hyperparameter	**Value / Setting**
Batch size	4 (per device)
Gradient accumulation	4
Block size	512
Epochs	20
Optimizer	AdamW
Learning rate	5e-4
Weight decay	0.001
Scheduler	Linear (with warm-up)
Warm-up steps	$\approx$8,900 (5%)
Total update steps	$\approx$177k
Gradient clipping	1.0
Loss function	Causal Loss

Table 6: Hyperparameters used during continual pretraining of UrduGPT and UrduQwen models.

Hyperparameter	**Value / Setting**
Batch size	4
Block size	512
Epochs	20
Optimizer	AdamW
UrduGPT lr	3e-5
UrduQwen lr	5e-6
Scheduler	Linear (with warm-up)
Warm-up steps	500
Loss function	Causal Loss

Table 7: Hyperparameters used during instruction tuning of UrduGPT and UrduQwen.

sourced from Pakistan's leading news outlets: Geo News and Dawn News, with the rest from other PEMRA-regulated media houses.[1]

[1] https://pemra.gov.pk/

To improve temporal and topical diversity, we scraped an additional 100,000 Urdu news articles from two highly credible PEMRA-regulated websites: Jang and Dawn News. Articles from Jang were collected between 2023 and 2025, and those from Dawn were from 2018 to 2023. These texts cover national/international affairs, lifestyle, entertainment, sports, and technology. We also incorporate 960 BBC Urdu to enhance factual and domain-specific coverage.

Preprocessing Most texts in the Urdu Wikipedia crawl contained markup and non-textual artifacts. To extract clean and coherent text, we performed several preprocessing steps: *(i)* remove image-related and location template content; *(ii)* strip markup elements such as tables, categories, and hyperlinks; *(iii)* discard style tags and other non-linguistic tokens; and *(iv)* filter out articles shorter than 200 characters. After applying these preprocessing steps, the resulting Wikipedia subset comprised approximately 135,000 clean articles suitable for pretraining.

A.5 Computational Cost

All experiments were conducted on four NVIDIA RTX A6000 (49 GB) GPUs using the `Accelerate` library for distributed training. We report approximate training time and computational cost to provide additional transparency regarding resource requirements.

Stage	MKGPT		MKQwen	
	Time (hrs)	GPU (hrs)	Time (hrs)	GPU (hrs)
Continual Pretraining	12	48	18	72
Instruction Tuning	4	16	3	12
Total	**16**	**64**	**21**	**84**

Table 8: Approximate training time and GPU usage for continual pretraining and instruction tuning.

A.6 LLM-as-a-Judge Prompt

The exact prompt used for LLM-as-a-judge evaluation with GPT-4o-mini is provided below.

```
You are an expert Urdu linguist and evaluator. Your task is to assess the quality of multiple model responses given an instruction and a gold reference answer.

Evaluate responses based on coherence, fluency, richness, and relevance.

The responses are anonymized and shown in random order. Do not assume
```

any response is from a stronger or weaker model. Evaluate each response independently.

Instruction: {instruction}
Gold Reference: {reference}
model1 Response: {response1}
model2 Response: {response2}
model3 Response: {response3}

Evaluate each response according to the following criteria:

1. Fluency and Coherence (0–40)
2. Instruction Relevance (0–30)
3. Vocabulary Richness (0–15)
4. Conceptual Correctness (0–15)

Provide the evaluation in the following JSON format:

```
{ "model1": {
"fluency_coherence": <0–40>,
"instruction_relevance": <0–30>,
"vocabulary_richness": <0–15>,
"conceptual_correctness": <0–15>
},
"model2": {...},
"model3": {...}
}
```