Markhor: A Scalable Pipeline for Training Urdu Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) have established state-of-the-art performance across a wide range of NLP tasks, yet their effectiveness in low-resource languages such as Urdu remains limited due to insufficient training data and suboptimal tokenization. To address this shortcoming, we propose *Markhor*[1], a comprehensive and scalable training pipeline for developing Urdu-specific LLMs, employing a knowledge distillation approach using two teacher models (DeepSeek and GPT-4o-mini). The pipeline spans data collection, tokenizer training, continual pretraining, and supervised instruction tuning, resulting in two models, MKGPT and MKQwen, based on GPT and Qwen architectures. A key finding is that initializing models from pretrained weights remains effective even when the target language employs a completely new tokenizer and vocabulary, with no lexical overlap with the source model. We evaluate these models on multiple classification tasks (hate speech detection, fake news detection, and emotion classification) and open-ended question answering. Experimental results demonstrate both MKGPT and MKQwen consistently outperform the baselines on all tasks, highlighting the effectiveness of our methodology in developing open-source Urdu LLMs. We release our dataset, models, and code at URL withheld.

1 Introduction

Large language models (LLMs) have redefined the state of the art across a wide range of natural language processing (NLP) tasks, enabling capabilities that were infeasible only a few years ago. However, their performance varies significantly across different languages, with most open-source multilingual LLMs showing subpar results in low-resource languages such as Urdu (Adeeba et al., 2025). Although open-source LLMs offer greater accessibility and control, they often underperform proprietary counterparts in terms of coverage, robustness, and reasoning depth. These limitations are particularly pronounced for low-resource settings. In the case of Urdu, insufficient representation in pretraining corpora leads to limited vocabulary coverage and weak tokenization performance, as highlighted by recent Urdu benchmark studies (Arif et al., 2024; Tahir et al., 2024).

As the national language of Pakistan[2], Urdu has 232 million speakers (International Center for Language Studies, n.d.) and a growing presence in social media communication (Ullah et al., 2024), underscoring the need to strengthen its representation in large language models, especially open-source LLMs. A promising approach is *knowledge distillation* (KD), which transfers linguistic competence from a strong *teacher* model to a smaller or less capable *student* model (Yuan et al., 2024). Within the domain of LLMs, KD has been used to *(i)* improve task effectiveness, *(ii)* compress models for efficiency, and *(iii)* enable self-improvement through synthetic supervision signals (Xu et al., 2024). However, simply fine-tuning existing open source checkpoints rarely bridges the gap for underrepresented languages; challenges such as data scarcity, domain drift, and cross-lingual transfer inefficiencies persist (Kadyrbek et al., 2025). In this work, we explore the paradigm of pretraining open-source, monolingual Urdu LLMs and jointly incorporating knowledge distillation signals during instruction tuning to enhance model capability and accessibility for a language that remains underrepresented in the current LLM ecosystem.

To this end, we propose *Markhor*, a comprehensive and scalable training pipeline encompassing data collection, tokenizer training, continual pretraining, and supervised instruction tuning via knowledge elicitation and distillation. Using this

[1] Named after the national animal of Pakistan

[2] https://en.wikipedia.org/wiki/Urdu

pipeline, we train two Urdu-specific LLMs based on two different architectures: GPT and Qwen (MoE), resulting in, *MKGPT* and *MKQwen*. For *knowledge elicitation* and *distillation*, we employ two strong teacher models, DeepSeek and GPT-4o-mini. Resulting models are evaluated on five publicly available classification datasets across three tasks (e.g., fake news detection, hate speech detection, and emotion classification). We additionally collect an open-ended question-answering benchmark spanning five domains to measure factual consistency and domain understanding. Experimental results demonstrate that our models, MKGPT and MKQwen, consistently outperform their respective baselines across all evaluation tasks. Our key contributions are summarized as follows:

- We collect, curate, and release a large-scale Urdu pretraining corpus, an instruction-following dataset, and an open-ended QA evaluation benchmark.
- We develop Urdu-specific tokenizers and perform continual pretraining with tokenizer adaptation, showing that pretrained models can be effectively adapted despite fully mismatched vocabularies.
- We introduce two Urdu-specific LLMs developed through end-to-end data collection, pretraining, instruction following, and comprehensive evaluation, based on two representative architectural families of GPT and Qwen. The strong performance of these models on Urdu benchmarks empirically demonstrates the effectiveness of our training paradigm for improving low-resource language representation in LLMs by integrating pretraining, knowledge elicitation, and distillation.

2 Related Works

This section reviews previous work on large language models in low-resource languages and Urdu, and knowledge distillation techniques for LLMs.

Urdu LLMs. Recent work has examined the performance of multilingual and general-purpose LLMs in low-resource languages, showing consistent performance gaps relative to high-resource languages. Arif et al. (2024) systematically evaluated general-purpose LLMs versus domain-specific fine-tuned XLM-RoBERTa and mT5 across seven Urdu tasks, demonstrating that the latter often surpasses LLMs such as GPT4-Turbo and Llama3-8B, even with advanced prompting techniques. Similarly, Tahir et al. (2024) benchmarked GPT-3.5-Turbo, Llama2, and BLOOMZ across 14 tasks and 15 Urdu datasets, showing that task-specific models and traditional baselines outperform general LLMs in zero-shot conditions. Kazi et al. (2025) further explored few-shot and chain-of-thought prompting for Urdu question answering, revealing sensitivity to question type and linguistic complexity, thereby underscoring challenges in Urdu reasoning. While Fiaz et al. (2025) introduced UrduLlama via continual pretraining and instruction tuning, the lack of public release restricts reproducibility. Collectively, these efforts highlight an active yet fragmented landscape of Urdu LLM research, with persisting gaps in standardized evaluation, model openness, and Urdu-centric pretraining.

Beyond Urdu, multilingual efforts such as BLOOMZ (Workshop et al., 2022), mGPT (Shliazhko et al., 2024), and Aya (Üstün et al., 2024) have broadened coverage for low-resource languages, yet Urdu remains underrepresented in pretraining corpora, tokenizers, and benchmarks.

Knowledge Distillation for LLMs. Knowledge distillation (KD: Hinton et al. (2015)) has emerged as a main strategy for compressing and transferring knowledge between large and small models. Early works such as DistilBERT (Sanh et al., 2019) and TinyBERT (Jiao et al., 2020) established task-agnostic and task-specific KD frameworks, respectively, focusing on encoder-based architectures.

In contrast, recent efforts have adapted KD for causal and generative models. Gou et al. (2021) provided a comprehensive overview of KD formulations. Yang et al. (2024) introduced self-distillation for instruction-tuned LLMs, leveraging synthetic supervision to enhance reasoning ability without labeled data. Similarly, Shridhar et al. (2023) proposed a teacher-student paradigm for efficient instruction tuning, reducing training costs while preserving generation quality. In multilingual contexts, Gupta et al. (2023) demonstrated that KD can effectively transfer cross-lingual knowledge from high-resource to low-resource languages, improving downstream performance even with limited target-language data. Other works such as Yuan et al. (2024) and Xu et al. (2024) have shown that self-distillation and multi-teacher strategies can stabilize LLM training and enhance linguistic generalization, reinforcing the viability of knowl-

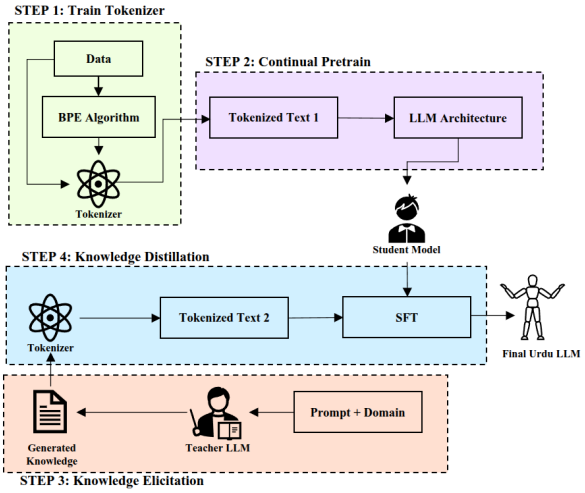

Figure 1: Four-stage Markhor pipeline for Urdu LLM training. **SFT**: Supervised Fine-Tuning. **BPE**: Byte-Pair Encoding.

edge distillation as a data- and compute-efficient alternative to full-scale pretraining.

Unlike prior studies that rely on fine-tuning or continual pretraining of multilingual backbones, our work introduces a pipeline that encompasses all stages of LLM training to train two monolingual Urdu causal LLMs. Compared to most previous distillation studies that target encoder-based or English-centric architectures, we integrate multi-teacher KD directly into the instruction tuning of GPT-based models. This dual focus on language specialization and distillation-driven efficiency represents a first step toward scalable, open, and reproducible Urdu LLM development under constrained compute budgets.

3 Building Urdu LLMs

Figure 1 presents the four-stage pipeline for developing an Urdu LLM. Following data collection, we first train an Urdu-specific tokenizer, which is then used for continual pretraining on base architectures to obtain Urdu-adapted student models. Next, knowledge is elicited from proprietary teacher models to generate instruction–response pairs, which are used for supervised instruction tuning of the student models, resulting in the final Urdu LLMs. Each stage is described in detail below.

3.1 Pretraining Data Collection

For continual pretraining, we curated a diverse Urdu corpus drawn from multiple publicly available sources, including encyclopedic, literary, and news domains, as shown in Table 1. The data covers both classical and contemporary Urdu.

More details about data sources and preprocessing are provided in Appendix A.4. The final pretraining data contains ∼1.3 million Urdu articles from all sources 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) issue in existing multilingual tokenizers. On a held-out Urdu corpus, GPT-4o and Llama3 tokenizers average ∼4 tokens per word, while GPT2 and Qwen3 tokenize at near character level. Given an average word length of ∼6 characters in the held-out corpus, such fine-grained segmentation risks losing meaningful linguistic cues.

Since the base architectures of GPT2 and Qwen3 employ the BPE algorithm, we trained new BPE tokenizers on the collected Urdu corpus. Before training, we performed Unicode-based normalization to unify visually similar Urdu characters, removed optional diacritics and elongation (Tatweel) marks, and standardized punctuation and digits.

To maintain compatibility with the embedding layers of the base models, we kept the vocabulary sizes identical to the original tokenizers: 50,257 for GPT2 and 151,643 for Qwen3, corresponding to roughly 49K and 150K merge operations, respectively. This alignment allowed us to retain the embedding and LM head dimensions without reinitializing weights. Table 2 shows the fertility of the newly trained tokenizers on the same held-out corpus. The results confirm a substantial reduction in average tokens per word, indicating improved subword segmentation and vocabulary coverage.

3.3 Continual Pretraining

In the second step, we tokenized the collected Urdu corpus using the tokenizer trained in Section 3.2, and then adapted the selected models under the causal language modeling (CLM) objective. This step aims to enable the models to internalize Urdu syntax, morphology, and lexical dependencies, thereby improving their representational understanding of the language. Our continual pretraining setup differs slightly from the conventional setting. Keeping the new tokenizer's vocabulary size identical to that of the original tokenizer maintained architectural compatibility, preserving the input embedding and LM-head dimensions, so the original weights could be reused. However, the new

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. $\hat{T}$: $\sim$ avg. token count, **UGPT:** UrduGPT, **UQwen:** UrduQwen.

Tokenizer	Avg. Fertility (tokens/word)
GPT-4o Tokenizer	4.12
Llama-3 Tokenizer	4.36
GPT2 (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)

tokenizer altered the token-to-ID mapping for these layers. Hence, our approach can be best described as *continual pretraining with tokenizer adaptation*, where pretrained transformer weights provide structural priors while token-level representations are relearned during further training.

Thus, continual pretraining in this work refers to additional training of a pretrained model on an Urdu-specific corpus while retaining its original parameters as initialization.

We employed two base architectures: GPT2 and Qwen3. For GPT2, three variants (175M, 335M, 774M) were evaluated, among which the 175M model converged most efficiently and exhibited stable validation loss, 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. Through empirical testing, a maximum sequence length of 512 was found effective for both models. The remaining hyperparameters are detailed in Section 4.1.

It is important to note that GPT2 has no prior exposure to the Urdu language, while Qwen3-0.6B includes limited Urdu coverage, as verified through qualitative probing of its zero-shot Urdu generation capability. The objectives of this stage were therefore to (i) improve next-token prediction and contextual completion in Urdu and (ii) provide a stronger initialization for the subsequent instruction-tuning stage. The resulting models after continual pretraining are named as *UrduGPT* and *UrduQwen*, derived from their base models.

Domain	Subdomain	Topic	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).

3.4 Knowledge Elicitation

In the third stage, we performed knowledge elicitation using three teacher models (GPT-4o-mini, DeepSeek, and Llama3-70B) to generate instruction-response pairs in Urdu. The objective was to obtain diverse and domain-rich samples covering science and general knowledge. Given a domain, we prompted ChatGPT to generate 1,000 domain-related topics, which were then expanded into subdomains to maximize topic diversity and minimize duplicate entries. After manual review and removal of redundant or overlapping items, 766 unique topics were retained, as shown in Table 3.

Each topic was dynamically incorporated into the prompt for producing instruction–response pairs. The teacher models were instructed to produce 50 pairs per topic in Urdu. Due to token length constraints, some responses contained incomplete JSON objects. We applied a post-processing step to filter and remove partially generated or invalid samples.

Quality Assurance We employed multiple quality control measures to ensure linguistic correctness and factual consistency. First, we manually reviewed a subset of samples from each teacher model to assess overall response quality. Llama3-70B occasionally produced mixed Urdu-Chinese text. GPT-4o-mini and DeepSeek generations are available in high-quality Urdu. Therefore, we excluded Llama3-70B from further use. Second, automatic filtering removed near-duplicate or low-content responses. Near-duplicate responses were identified using token-level lexical overlap, computed via Jaccard similarity; responses with a similarity score greater than 0.9 were removed. Low-content responses were defined as outputs containing fewer than five words. Third, outputs from the two teacher models were cross-compared during early generation phases to identify and discard inconsistent or clearly incorrect samples. While large-scale factual verification was infeasible, these procedures collectively ensured a linguistically robust and factually plausible dataset suitable for instruction tuning. The final dataset contains 103,463 instruction-response pairs. Table 3 presents the topic- and subdomain-wise distribution of generated pairs by each teacher model. GPT-4o-mini produced a larger share of the instruction data due to its support for batch API generation, whereas DeepSeek relied on single-request calls, resulting in lower throughput.

3.5 Knowledge Distillation

The final stage is instruction tuning to distill the knowledge of teacher models into the student models via supervised finetuning. Two separate pipelines were developed for UrduGPT and UrduQwen due to differences in their base architectures and chat formatting schemes.

UrduGPT lacks a built-in chat template, so each instruction-response was flattened into plain text as: `Instruction:{instruction}\nResponse:{response}{eos_token}`. This ensured that the model explicitly learned the mapping between user instructions and corresponding responses. Instruction tuning was implemented in two stages: first, UrduGPT distilled DeepSeek-generated data containing relatively simple instruction-response pairs; subsequently, GPT-4o-mini data was used to transfer more complex, domain-specific knowledge. This staged approach allowed the model to gradually progress from general to specialized understanding, improving convergence stability.

UrduQwen Each instruction–response pair applied Qwen3's predefined chat template by calling `apply_chat_template()` function. The messages followed a three-turn system-user-assistant structure, ensuring compatibility with the model's conversational alignment. Unlike UrduGPT, staged training was not applied to UrduQwen because of its larger capacity and faster convergence observed during pretraining. Therefore, all instruction-tuning data from both teacher models was provided jointly.

In both cases, the models were optimized to minimize next-token prediction loss over instruction-response pairs, effectively distilling linguistic and task knowledge from the teacher models into the student models. The maximum sequence length was set to 512 tokens for both models.

4 Experiments

4.1 Experimental Setup

Pretraining Setup Both UrduGPT and UrduQwen were initialized from their respective base model weights and trained using mixed-precision arithmetic (FP16/FP32). All parameters in both models, including embedding, multi-head attention, and LM head layers, were optimized end-to-end during pretraining. The AdamW optimizer was used with a learning rate of 5e-4, linear decay scheduling, and gradient accumulation over four steps. Warmup steps were set to 5% of the total training steps to stabilize early optimization. Both models were trained for 20 epochs on four NVIDIA RTX A6000 (49 GB) GPUs using the `Accelerate` library for distributed training.

Using their respective tokenizers, the Urdu corpus was tokenized into approximately 145 million tokens for UrduGPT and 140 million for UrduQwen. The corpus was split into 90% training and 10% validation subsets. Due to the slightly smaller number of tokens, UrduQwen had a marginally lower total number of training and warm-up steps. A single checkpoint was maintained and updated throughout training based on the lowest validation loss. Key hyperparameters, chosen through a series of empirical experiments, are listed in Table 6 (Appendix A.3).

Figure 2 compares the training and validation loss curves of UrduGPT and UrduQwen during continual pretraining. A common trend observed in both models is that the validation loss remains

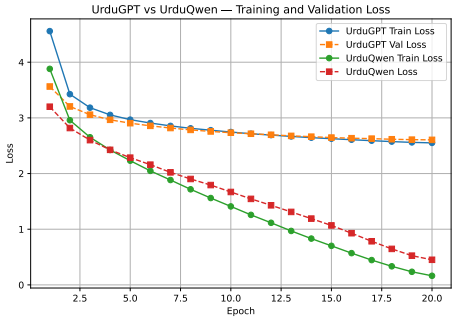

Figure 2: Pretraining learning curve of UrduGPT / UrduQwen.

lower than the training loss during the initial epochs, which typically indicates effective regularization and under-fitting at early stages. As training progresses, the training loss gradually falls below the validation loss — around the 11th epoch for UrduGPT and much earlier for UrduQwen, suggesting the models begin fitting more closely to the training data. Overall, both models exhibit a smooth and consistent decline in loss, reflecting stable convergence. However, UrduQwen demonstrates a substantially faster rate of loss reduction and achieves lower overall values for both training and validation loss. This indicates a more efficient learning process, likely attributed to its larger model capacity and architectural improvements over UrduGPT.

Knowledge Distillation Setup We applied supervised fine-tuning (SFT) to UrduGPT and UrduQwen, distilling knowledge from teacher models to equip two student models with instruction-following capabilities. Instruction tuning was carried out in two stages for UrduGPT, first on DeepSeek-generated data and then on GPT-4o-mini data. UrduQwen was finetuned 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 the learning rate for UrduQwen, which was 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), evaluating after each epoch and retaining the checkpoint with the lowest validation loss. We refer to the final distilled models as *MKGPT* and *MKQwen*.

Figure 3 presents the instruction-tuning learning curves for both models. For MKGPT, the loss

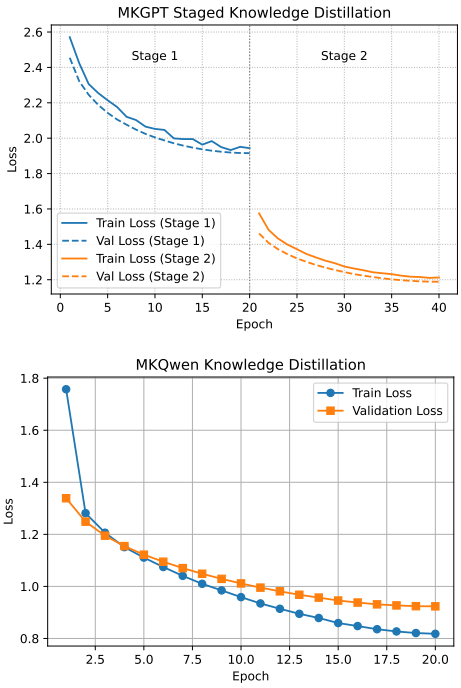

Figure 3: Instruction-tuning learning curves of MKGPT (top) and MKQwen (bottom).

curves in both stages showed classic textbook behavior: a sharp initial drop followed by gradual stabilization, indicating well-configured hyperparameters and stable convergence. The validation loss curves remained slightly below the training loss curves, suggesting effective generalization. The final validation loss reached just below 1.2. In contrast, MKQwen's curves show smooth convergence, but the validation loss exceeds the training loss in later epochs, implying mild overfitting as the model began fitting the training data more closely. However, the validation trend closely followed the training curve, with final losses around 0.9-0.8, reflecting strong overall learning stability.

4.2 Baselines

The original GPT2 and Qwen3 base models serve as baselines. We compare the performance of MKGPT and MKQwen with their respective base versions across four downstream tasks: hate speech detection, fake news detection, emotion classification, and question answering. These baselines provide a reference to evaluate the effectiveness of the proposed Markhor training pipeline, which includes new tokenizer training, continual pretraining, and knowledge distillation, and to assess the generalization of our models across tasks.

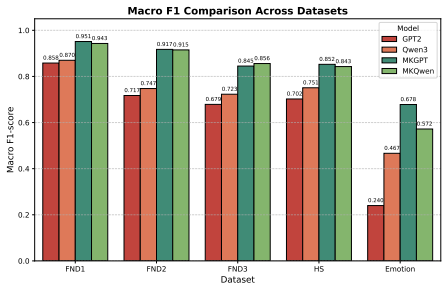

Figure 4: Macro-F1 across five classification datasets. **FND**: Fake News Dataset, **HS**: Hate Speech.

4.3 Evaluation

Datasets For downstream evaluation, we use five publicly available Urdu datasets, including one dataset 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 curated a question answering dataset across five domains (i.e., physics, chemistry, biology, history, and geography) on which the models were trained, by instructing a native speaker to create 100 QA pairs per domain, with responses of at least 2–3 sentences, allowing for diverse topics within each domain. We evaluated on both *(i)* in-distribution examples — a held-out test split comprising 100 QA pairs per domain sampled from the instruction-tuning dataset, to examine knowledge capture, and *(ii)* out-of-distribution examples — an outsourced human-annotated dataset of equal size to assess generalization and factuality.

Metrics For classification tasks, we report standard metrics including precision, recall, F1-score, and overall accuracy. To assess the generation quality of these Urdu LLMs, we report BERTScore-F1 to measure semantic similarity with gold responses. In addition, GPT-4o-mini is used as a judge to rate each model's coherence, instruction adherence, vocabulary richness, and conceptual accuracy, assigning an overall score out of 100 for each response.

Downstream Task Adaptation For the classification tasks, we fine-tuned our models using the `AutoModelForSequenceClassification` class from Hugging Face. Each dataset was split into 68% training, 12% validation, and 20% test sets. The maximum sequence length was selected based on the token length distribution of each dataset. All models were trained for five epochs with a learning rate varied between 1e-5 and 1e-6, selecting the configuration that conceded the lowest validation loss. The `load_best_model_at_end` option was enabled to ensure optimal checkpoint selection.

For open-ended question answering, we used `model.generate()` function with nucleus sampling (p=0.9), top-k sampling (k=50), temperature of 0.8, and a repetition penalty of 1.2, generating up to 200 new tokens per prompt. These settings aimed to balance coherence and diversity in generated responses while preventing repetition.

4.4 Results and Discussion

Classification Tasks Figure 4 presents the macro-F1 results of the baseline and distilled Urdu language models across five classification datasets.

Both MKGPT and MKQwen consistently outperform their vanilla counterparts, demonstrating the effectiveness of Urdu-specific pretraining and knowledge distillation. Among the two, MKGPT shows slightly better overall performance, which can be attributed to its smaller parameter size being more suitable for scenarios with limited training data. Given that the evaluation datasets used contain only a few thousand examples each, the smaller MKGPT architecture may have achieved better generalization, whereas the larger MKQwen model likely required more data to fully leverage its capacity. When comparing the base models, Qwen3 performs better than GPT2, likely due to Urdu being included in its multilingual pretraining. Interestingly, in fake news detection (FND1-FND3), models achieve higher performance on the short-text dataset (FND1) compared to the long-article datasets (FND2, FND3), as shorter texts have higher information density. In contrast, the models show the weakest performance on the Emotion dataset, which can be attributed to the fact that this dataset is the smallest (around 1000 examples) and has higher label complexity with six emotion categories. Precision, recall, and accuracy trends follow similar patterns and are provided in the Appendix due to space constraints.

Question-Answering Figure 5 compares the performance of MKGPT and MKQwen with the Qwen3 baseline on the question-answering task. GPT2 is excluded since it is not inherently trained for chat or QA. Across both evaluation metrics, the newly trained Urdu models consistently outperform the Qwen3 baseline. Across five domains, Qwen3 exhibits nearly constant BERTScore F1 ranging

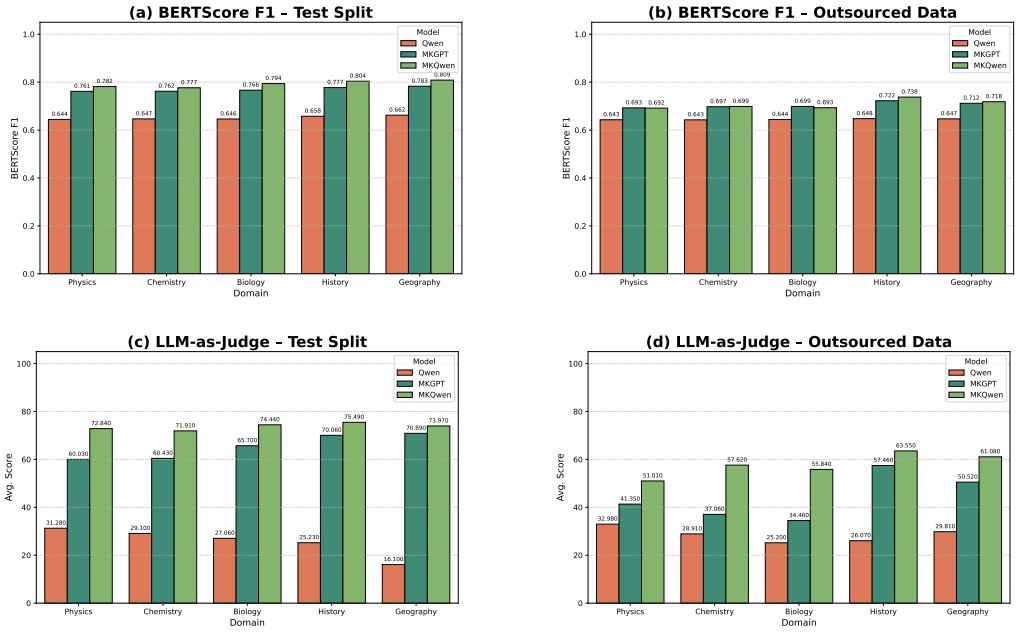

Figure 5: Question-answering evaluation results across domains. (a–b) BERTScore F1 on test and outsourced datasets. (c–d) LLM-as-Judge scores (aggregated out of 100) on test and outsourced datasets.

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 tokens in the gold answers and generated responses, potentially awarding high scores to responses with correct wording but grammatically incorrect outputs, whereas LLM-as-judge evaluates coherence, instruction adherence, vocabulary richness, and conceptual accuracy.

MKQwen generally outperforms MKGPT across all domains and splits, likely due to Qwen3's larger model size, pre-existing chat-oriented architecture, Urdu language pretraining and instruction-following capabilities, providing stronger bases for our 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 scarce science-related data. Science knowledge is primarily transferred to the models via distillation. Performance on outsourced 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

In this study, we introduce Markhor, a scalable pipeline for training Urdu LLMs, and use it to develop two models, MKGPT and MKQwen, based on GPT2 and Qwen3. The models are pretrained with a new Urdu-specific tokenizer and corpus, and tuned via knowledge distillation from DeepSeek and GPT-4o-mini. The models were evaluated on three classification tasks across five datasets and two question answering test sets. The results show that both models substantially outperform baseline models across all datasets. MKGPT excels in classification, while MKQwen shows superior performance on QA. The superior results demonstrate the effectiveness of our training pipeline.

We find that initializing a new language model with pretrained weights from a model using a completely different tokenizer and vocabulary can still substantially improve learning, even when the embedding layer is fully mismatched. Our experiments demonstrate the feasibility of pretraining a low-resource monolingual language model with a limited corpus by training a new tokenizer, resetting the vocabulary, and initializing weights from a strong pretrained base model, despite the base model having no prior exposure to the target language and containing no tokens from that language in its vocabulary. For future work, we plan to experiment with larger models, include reasoning-focused domains, explore joint Urdu-Hindi training, and incorporate alignment with human feedback to further enhance model capabilities.

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.

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.

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.

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.

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*.

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*.

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

International Center for Language Studies. n.d. Most spoken languages in the world. https://www.icls.edu/blog/most-spoken-languages-in-the-world. Accessed: 2025-01-16.

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.

Nurgali Kadyrbek, Zhanseit Tuimebayev, Madina Mansurova, and Vítor Viegas. 2025. The development of small-scale language models for low-resource languages, with a focus on kazakh and direct preference optimization. *Big Data and Cognitive Computing*, 9(5):137.

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.

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*.

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*.

Ashraf Ullah, Khair Ullah Khan, Aurangzeb Khan, Sheikh Tahir Bakhsh, Atta Ur Rahman, Sajida Akbar, and Bibi Saqia. 2024. Threatening language detection from urdu data with deep sequential model. *Plos one*, 19(6):e0290915.

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*.

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.

Dataset	Model	Acc	Pre	Rec	F1
FND1	GPT2	85.8	86.0	85.8	85.8
	Qwen3	87.0	87.0	87.0	87.0
	MKGPT	95.1	95.2	95.1	95.1
	MKQwen	94.3	94.3	94.3	94.3
FND2	GPT2	71.8	71.7	71.7	71.7
	Qwen3	74.8	74.8	74.8	74.7
	MKGPT	91.8	91.8	91.7	91.7
	MKQwen	91.5	91.6	91.6	91.5
FND3	GPT2	70.8	71.6	67.8	67.9
	Qwen3	73.1	72.4	72.2	72.3
	MKGPT	85.0	84.9	84.2	84.5
	MKQwen	85.8	85.4	86.1	85.6
HS	GPT2	71.7	70.5	70.1	70.2
	Qwen3	76.7	75.9	74.6	75.1
	MKGPT	86.0	85.7	84.9	85.2
	MKQwen	85.1	84.6	84.0	84.3
Emotion	GPT2	28.9	35.9	26.5	24.0
	Qwen3	49.8	53.6	47.1	46.7
	MKGPT	67.1	68.4	67.8	67.8
	MKQwen	56.4	63.1	57.3	57.2

Table 5: Classification macro precision (Pre), recall (Rec), F1 and accuracy (Acc) by % across datasets. **FND**: Fake News Dataset, **HS**: Hate Speech.

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.

Hyperparameter	Value / Setting
Batch size	4
Block size	512
Epochs	20
Optimizer	AdamW
Learning rate	5e-4
Weight decay	0.001
Scheduler	Linear (with warm-up)
Warm-up steps	≈8,900 (5% of total steps)
Total training steps	≈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.

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 sourced from Pakistan's leading news outlets: Geo News and Dawn News, with the rest from other PEMRA-regulated media houses.[3]

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

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

were collected between 2023 and 2025, and those from Dawn were from 2018 to 2023. These texts cover national and international affairs, lifestyle, entertainment, sports, and technology. Furthermore, we incorporated 960 science articles from 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.