DunbaaBERT: From Sacrifice to Semantics

Iffat Maab[1,†] Waleed Jamil[2], Raphael Schmitt[3,4,†],

[1]Research and Development Center for Large Language Models (LLMC), National Institute of Informatics, Tokyo, Japan, [2]Independent Researcher, Edinburgh, United Kingdom, [3]School of Computation, Information and Technology, Technical University of Munich, Germany, [4]Institute of General Practice, Faculty of Medicine and Medical Center, University of Freiburg, Germany

[†]authors contributed equally

Correspondence: raphael.schmitt@uniklinik-freiburg.de

Abstract

Large language models have achieved strong performance across many NLP tasks, yet Urdu remains comparatively underexplored due to limited resources and fragmented evaluation settings. To address this gap, we introduce DunbaaBERT, a family of Urdu RoBERTa-base models trained from scratch with Byte-BPE vocabularies of 32k, 52k, and 96k tokens on a deduplicated 17 GB Urdu corpus. We evaluate DunbaaBERT across intrinsic and downstream Urdu NLP benchmarks covering linguistic acceptability, news classification, offensive language detection, and sentiment analysis, while analyzing vocabulary-size effects on performance, pre-training dynamics, and computational efficiency. Across benchmarks, the DunbaaBERT variants achieve competitive performance against strong multilingual baselines while consistently maintaining favorable efficiency trade-offs. Under our controlled experimental setup, increasing the vocabulary size does not consistently improve downstream effectiveness, with DunbaaBERT$_{32k}$ repeatedly providing the strongest overall efficiency profile. Overall, our results demonstrate that carefully curated Urdu-specific encoder models can remain highly competitive despite comparatively compact model and training scales. All models are released under the MIT license.

1 Introduction

Large language models have demonstrated strong performance across many NLP tasks through zero-shot, few-shot, and in-context learning approaches, often reducing the need for task-specific fine-tuning (Brown et al., 2020; Kojima et al., 2022; Meta, 2024; Achiam et al., 2023; Touvron et al., 2023). However, the benefits of these methods are not distributed equally across languages. Their effectiveness is most evident in high-resource languages such as English, where large-scale pretraining data and evaluation resources are widely available. In contrast, under-resourced languages such as Urdu remain comparatively underexplored and continue to face challenges in both language understanding and generation due to limited resources (Maab et al., 2026) and rich morphological structure (Blasi et al., 2021; Arif et al., 2024a). As a result, task-specific fine-tuning continues to play a central role in Urdu NLP, with prior work relying on supervised models for sentiment and emotion classification (Raza et al., 2025; Khan et al., 2026), hate-speech detection (Arshad et al., 2023; Bilal et al., 2023; Maab et al., 2026), named entity recognition (Zafar et al., 2025), AI-generated text detection (Al-Qasem, 2026), fake news detection (Dhiman et al., 2024), and more recently fact-checking (Ahmad et al., 2025).

Nevertheless, compared to higher-resource languages, Urdu still lacks a broad ecosystem of openly available and systematically evaluated encoder-only language models, depsite being spoken by over 100 million people (Anabtawi, 2026). In Pakistan, Urdu remains a key language of daily communication and cultural expression, even though English continues to hold official status (Rahman, 1997). Standard Urdu is written in an abjad script historically linked to the Persian and Arabic writing traditions (Alam and Hussain, 2022; Ubul et al., 2017). However, online and informal communication frequently appears in Roman Urdu, where Latin characters are used to represent Urdu text (Shakeel and Karim, 2020). In addition, bilingual speakers commonly alternate between Urdu and English, producing code-switched text across multiple linguistic units (Ansari et al., 2020; Bashir, 2011; Garcia, 2011). From an NLP perspective, progress in Urdu is further constrained by the scarcity and fragmentation of available resources. Existing datasets often vary in script, format, annotation scheme, and domain, spanning standard Urdu, Roman Urdu, and code-switched text (Saeed et al., 2021). Moreover, many studies use differ-

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 2074–2093
November 6-10, 2026 ©2026 Association for Computational Linguistics

ent training and evaluation setups, which make fair comparison across methods difficult. The absence of widely adopted benchmark datasets therefore remains a major obstacle to reliable and reproducible progress in Urdu NLP (Maab et al., 2026).

Recent efforts have partially improved Urdu language coverage through large-scale language-specific encoder models released within multilingual initiatives such as HPLT-BERT (Samuel et al., 2023; de Gibert et al., 2024). However, systematic analyses of Urdu-specific encoder models remain limited, particularly with respect to tokenizer design and vocabulary size. Prior work has shown that tokenization choices can substantially influence model quality in morphologically rich languages (Toraman et al., 2023), yet such analyses remain largely unexplored for Urdu. In addition, RoBERTa-based architectures have recently demonstrated strong performance-efficiency trade-offs in Portuguese NLP (Scheible-Schmitt et al., 2025), motivating their investigation in the context of Urdu encoder models.

To address these limitations, we introduce DunbaaBERT, a set of Urdu RoBERTa-base models trained with Byte-BPE vocabularies of 32k, 52k, and 96k tokens. We further provide intrinsic and downstream evaluations across multiple Urdu NLP benchmarks and linguistic probing tasks, while analyzing the impact of vocabulary size on model performance, training dynamics, and efficiency trade-offs. All models are released under the MIT license to support future research in Urdu NLP.

2 Related Works

Multilingual pretrained encoders have played a central role in extending Transformer-based language modeling to lower-resource languages such as Urdu. mBERT (Devlin et al., 2019) established a strong early multilingual baseline by training a single masked language model on Wikipedia corpora from multiple languages. XLM-R (Conneau et al., 2020) subsequently improved multilingual encoder pre-training by replacing Wikipedia-only supervision with substantially larger Common Crawl-derived multilingual data. More recently, mmBERT (Marone et al., 2025) has further increased multilingual scale, with the model family reported to cover over 1,800 languages and more than 3T training tokens, thereby including Urdu in a massively multilingual pre-training regime. By contrast, EuroBERT (Boizard et al., 2025) is explicitly designed for a restricted set of 15 European languages and is therefore not directly relevant as an Urdu baseline. While such multilingual encoders are highly important comparison points, they necessarily distribute modeling capacity across many languages and thus differ substantially from monolingual Urdu-focused pre-training.

Recent work on Urdu representation learning remains comparatively limited. One potentially relevant Hugging Face model `wajidhussain/bert-base-urdu`, is difficult to interpret reliably as an Urdu BERT baseline, since its public model card is inconsistent: it is tagged for text generation and includes GPT-2-style descriptive content rather than a standard encoder-style masked-language-model. We therefore treat it with caution and do not rely on it as a reference point.

Additional Urdu-language models hosted on Hugging Face likewise suffer from limited documentation and unclear training characteristics. The `mahwizzzz/UrduBert` model is presented as a RoBERTa-style encoder, but its model card does not provide sufficient details regarding training data, optimization setup, or pre-training duration. Preliminary experiments further suggested that the model may be substantially under-trained. Similar limitations apply to `mahwizzzz/Urdu-Bert`, whose architecture and training configuration are insufficiently documented and whose downstream behavior likewise indicates possible under-training.

Another publicly available checkpoint, `eshaaftab900/urdu-bert-base`, is described as a BERT-base model, but its model card provides only minimal technical information and lacks reproducibility-relevant details concerning corpus composition, tokenizer construction, and pre-training methodology. Due to these documentation limitations and the resulting difficulty for reliable scientific interpretation, we exclude these models from our primary comparative evaluation.

A clearer comparison point is provided by the HPLT Urdu model HPLT-BERT$_{ur}$[1], which is explicitly described as an encoder-only monolingual masked language model based on LTG-BERT, i.e., a BERT-family architecture adapted for large-scale monolingual pre-training (Samuel et al., 2023). Its underlying data originate from the HPLT corpus collection (de Gibert et al., 2024), which was constructed from large web-derived resources includ-

[1] https://huggingface.co/HPLT/hplt_bert_base_ur

ing Common Crawl and Internet Archive data, with document-level deduplication applied during corpus preparation. This makes HPLT-BERT$_{ur}$ a relevant recent baseline for Urdu encoder pre-training on large-scale crawl-based text.

Another noteworthy line of work is RUBERT, introduced by Khalid et al. (2021), which targets Roman Urdu rather than standard Urdu written in Perso-Arabic script. RUBERT was obtained by additional pre-training of English BERT on a large Roman Urdu Twitter corpus comprising 54 million tokens and 3 million sentences. While this work demonstrates the feasibility of Urdu-adjacent pre-training in noisy social media settings, its script, domain, and initialization strategy differ substantially from monolingual encoder pre-training for standard Urdu text. For this reason, we do not include RUBERT in our empirical comparison, as our study focuses on encoder models trained directly on Perso-Arabic Urdu corpora under controlled pre-training conditions.

3 Methods

3.1 Corpora

Our Urdu pre-training corpus was constructed from a combination of large-scale web data, encyclopedic text, and a filtered auxiliary parallel data source. The overall objective was to balance corpus size, stylistic diversity, and data quality while avoiding unnecessary complexity. To this end, we combined several complementary sources and applied targeted filtering only where the risk of noise was deemed particularly high.

3.1.1 Core Web Corpora

The core of our corpus consists of large-scale web-derived Urdu text. As principal sources, we used mC4 and several OSCAR subsets from CulturaX (Nguyen et al., 2023). These corpora provide broad topical and stylistic coverage and constitute the main source of linguistic variability in the final training mixture. While CulturaX represents a comparatively clean and pre-filtered corpus component, the datasets most likely still contain residual web-crawling artifacts such as boilerplate, mixed-language fragments, duplicated content, and malformed lines. Rather than aggressively filtering these sources toward a narrow stylistic target, we retained them largely in their original form to preserve domain diversity.

3.1.2 Wikipedia

To complement the web-derived sources with a comparatively clean and well-formed reference corpus, we included Urdu Wikipedia. The dump was converted into a one-document-per-line format, such that each line corresponds to one Wikipedia article. Compared to the web-crawled corpora, this source is more homogeneous in style and generally cleaner in terms of structure and language quality. We therefore used Wikipedia not only as an additional corpus component, but also as a reference distribution for unsupervised filtering of noisy auxiliary data. The corpus was created using WikiExtractor[2].

3.1.3 Auxiliary Data from NLLB

In addition to the core corpora, we considered Urdu text extracted from NLLB-derived parallel data (Schwenk et al., 2020; Fan et al., 2020) as an auxiliary source which we retrieved from OPUS (Tiedemann, 2012). This material introduces additional diversity and potentially increases the range of lexical and syntactic constructions present in the training corpus. However, in contrast to the core corpora, this source was treated as comparatively noisy, since automatically mined multilingual or parallel resources may contain language mixing, segmentation artifacts, metadata fragments, or other irregularities. For this reason, we did not incorporate it directly, but subjected it to an additional unsupervised filtering step before inclusion.

3.1.4 Unsupervised Filtering of Noisy Auxiliary Data

To reduce noise in the auxiliary NLLB-derived Urdu data without relying on manual annotation, we applied an unsupervised filtering procedure based on distributional similarity to a clean reference corpus. As reference material, we used Urdu Wikipedia, which was converted into a one-document-per-line format and used to train an unsupervised fastTex skip-gram model with subword information (Bojanowski et al., 2016). The resulting embedding space serves as a proxy for well-formed Urdu text.

Let $\mathcal{W} = \{w_1, \ldots, w_M\}$ denote a sample of reference lines from Urdu Wikipedia. In our experiments, we set $M = 50{,}000$, obtained from the preprocessed Wikipedia corpus in one-document-per-line format. For each $w_i \in \mathcal{W}$, we compute a

[2]https://github.com/attardi/wikiextractor

sentence representation $\mathbf{s}_i \in \mathbb{R}^d$ using the fastText model. The reference centroid is then defined as

$$\mathbf{r} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{s}_i.$$

Given a candidate line x, we compute its sentence representation $\mathbf{v}(x) \in \mathbb{R}^d$ and measure its similarity to the reference centroid using cosine similarity:

$$\mathrm{sim}(x) = \frac{\mathbf{v}(x)^\top \mathbf{r}}{\|\mathbf{v}(x)\| \, \|\mathbf{r}\|}.$$

In addition, we define a non-negative heuristic penalty term $P(x) \geq 0$, which captures surface-level indicators of noisy text. Concretely, $P(x)$ is implemented as an additive penalty based on simple character-level statistics, including (i) short sequence length, (ii) low proportion of Arabic script characters, (iii) high proportion of Latin characters, digits, or punctuation, (iv) leading punctuation, (v) repeated symbols, and (vi) the presence of common boilerplate patterns. Each condition contributes a fixed penalty increment, resulting in values typically in the range $[0, 1.9]$.

The final score is given by

$$F(x) = \mathrm{sim}(x) - P(x).$$

A line is retained if $F(x) \geq \tau$, where τ is a threshold chosen conservatively such that only clear outliers are removed. We set $\tau = 0.1$.

Applied to the NLLB-derived Urdu subset, this procedure processed a total of 18,566,353 lines, of which 17,189,655 were retained and 1,376,698 were discarded. Thus, the filtering step removed approximately 7.4% of the candidate lines while preserving the large majority of the auxiliary material. This outcome is consistent with our goal of applying a conservative filter that removes clear outliers without collapsing the stylistic diversity of the retained text.

This approach should not be interpreted as a strict language identification system. Rather, it provides an unsupervised estimate of how closely a candidate line resembles the distribution of clean Urdu reference text. To preserve stylistic and domain diversity, we applied this filtering step only to the noisy auxiliary data rather than to the full pre-training corpus.

Appendix B, Table 5 illustrates representative high- and low-ranked candidates according to the proposed heuristic scoring function, contrasting clean sentence candidates with boilerplate and non-linguistic artifacts.

Corpus	Size
mC4	17.0 GB
OSCAR-2019	869 MB
OSCAR-2109	604 MB
OSCAR-2201	344 MB
OSCAR-2301	982 MB
Urdu Wikipedia	364 MB
NLLB Urdu (filtered)	2.1 GB
NLLB Urdu (raw)	*2.2 GB*
Total before exact deduplication	**22.3 GB**
Final corpus after exact deduplication	**17.0 GB**

Table 1: Composition of the Urdu pre-training corpus. Sizes refer to the plain-text corpus files used during corpus construction. For NLLB, we report both the raw extracted Urdu subset and the retained subset after unsupervised filtering. The total before exact deduplication refers to the merged corpus after filtering, including the filtered NLLB portion.

3.1.5 Corpus Composition and Final Deduplication

Table 1 summarizes the corpus components used in our experiments. The corpus primarily consists of large-scale web-derived Urdu text from mC4 and multiple OSCAR/CulturaX subsets, complemented by Urdu Wikipedia and filtered Urdu material extracted from NLLB. Together, these sources provide broad linguistic and topical diversity while balancing scale and corpus quality.

Before final deduplication, the combined corpus amounted to approximately 22.3 GB of text. After filtering the auxiliary NLLB-derived material and merging all retained sources, we applied exact line-level deduplication, resulting in a final corpus size of approximately 17 GB. This step removes literal duplicates across sources while preserving most of the variability of the training data. We deliberately refrained from more aggressive fuzzy or near-duplicate removal procedures, since such methods may over-regularize the resulting corpus.

In the final corpus, each line corresponds to one document. After deduplication, the corpus was randomly shuffled prior to binarization and pre-training to avoid source-local ordering effects.

3.2 Pre-training

DunbaaBERT was trained from scratch using our modular pre-training framework introduced in Schmitt (2026), which provides a reproducible pipeline for large-scale RoBERTa-style model train-

ing and orchestrates binary dataset generation, distributed training, check-pointing, and experiment management. The framework builds on fairseq (Ott et al., 2019) as the underlying training engine.

All models followed the RoBERTa-base architecture and were trained for 100k update steps using dynamic masking with whole word masking (WWM), fixed input sequences of 512 tokens, and optimization settings closely following Liu et al. (2019). The use of WWM ensures that masking operates at the word level rather than on individual subword units, which is particularly relevant in the context of varying vocabulary sizes. A warmup phase of 10k updates gradually increased the learning rate to a peak of 7×10^{-4}, followed by polynomial decay to zero.

Training was conducted in mixed precision (FP16) with an effective batch size of 8k. Pre-training was performed on NVIDIA H100 GPUs (96GB), using two GPUs for DunbaaBERT$_{32k}$ and DunbaaBERT$_{96k}$, and four GPUs for DunbaaBERT$_{52k}$. Aside from the vocabulary sizes under investigation, all pre-training settings were kept identical across model variants to isolate tokenizer effects.

During pre-training, sequences were constructed while preserving natural sentence boundaries, avoiding mid-sentence truncation and maintaining linguistic structure throughout optimization.

4 Experiments

4.1 Datasets

Linguistic Acceptability To assess fine-grained grammatical knowledge, we include evaluation on **URBLIMP** (Adeeba et al., 2025), a benchmark of minimal pairs designed to probe syntactic and morphosyntactic competence in Urdu. URBLIMP comprises 5,696 minimal sentence pairs targeting ten core linguistic phenomena, including agreement, argument structure, and word order variations. Each pair contrasts a grammatical and an ungrammatical sentence differing only minimally. For our analysis, we report results on 19 fine-grained evaluation categories derived from these phenomena, enabling a more detailed assessment of model behavior across specific linguistic constructions. Models are evaluated following the BLiMP protocol (Warstadt et al., 2020), i.e., by assigning higher probability to the grammatical sentence in each pair. We report accuracy for each category and use the macro-average across all categories as the overall URBLIMP score, complementing downstream tasks by evaluating fine-grained grammatical knowledge.

News Domain Classification **COUNT19** contains 10,451 documents collected from major Urdu news websites, including Geo, ARY, Express, Samaa, and BBC Urdu (Hamza et al., 2019). It covers seven news domains: International, National, Science, Sports, Health, Business, and Entertainment. The dataset is relatively balanced across categories, with each class containing approximately 1,480–1,509 documents, and contains 3.19M tokens, 91,840 unique unigram tokens, and 759,997 unique bigram tokens. Following the original dataset construction by Hamza et al. (2019), for COUNT19, we used the concatenation of the news title and article content as the input text for multi-classification among the seven categories.

Offensive Language Detection **USADC** is an Urdu-script abusive language detection dataset formulated as a binary classification task utilized by Maab et al. (2026); Arif et al. (2024b); Amjad et al. (2022). The dataset contains 5,670 instances, with 2,816 labeled as *Normal* and 2,854 as *Abusive*. Unlike most Roman Urdu abuse detection datasets (Maab et al., 2026), USADC preserves the native Urdu script, making it particularly useful for evaluating models on script-specific lexical, morphological, and orthographic patterns in Urdu abusive language detection.

Sentiment Classification Since sentiment classification is one of the most widely studied downstream tasks in Urdu NLP (Ashraf et al., 2024; Khan et al., 2022; Ashraf et al., 2023), we include it as an important benchmark task. For this task, we use **PSL–Kabaddi** benchmark, an Urdu Twitter sentiment dataset derived from a collection of sports-related tweets about the Pakistan Super League (PSL) and Kabaddi (Maqsood, 2023). The original data were collected from trending topics in Pakistan and subsequently filtered to remove non-Urdu tweets. The benchmark contains tweet-level aspect and sentiment annotations. In our experiments, we use the sentiment polarity labels for classification. The dataset contains 2,924 sentiment labels, with 1,577 positive, 1,100 negative, and 245 neutral labels, indicating a moderately imbalanced sentiment distribution.

We also use an Urdu **IMDB** movie review dataset for document-level binary sentiment classi-

fication task[3]. This dataset is an Urdu translation of the original English IMDB movie review corpus, produced using Google Translate, and contains 50,000 reviews evenly split between positive and negative classes; it has also been used in prior Urdu sentiment analysis studies (Irum and Tahir, 2025; Arif et al., 2024b; Hassan et al., 2024).

4.2 Setup

All downstream experiments were conducted using a unified fine-tuning setup based on PyTorch and HuggingFace Transformers. We performed hyperparameter optimization (HPO) across multiple learning rates, batch sizes, and random seeds, training each configuration for up to 30 epochs with early stopping (patience = 3). Best-performing configurations were selected based on validation macro-F1 and subsequently evaluated on the test set using a fixed evaluation batch size of 8. Additional implementation details, hyperparameter ranges, hardware specifications, dataset splits, and baseline properties are provided in Appendix A. To characterize variability across the complete hyperparameter search space, Appendix D.3, Table 7, reports results for all 60 configurations per model and benchmark, comprising five learning rates, four batch sizes, and three random seeds. Computational costs are summarized in Appendix E.

4.3 Evaluation and Efficiency Metrics

We report Accuracy, Macro-F1, and normalized efficiency (Norm. Eff.) for all downstream tasks. Within each benchmark, normalized efficiency is defined as

$$\text{Norm. Eff.} = \left(\frac{\text{Macro-F1}}{100}\right) \times \left(\frac{\text{SPS}}{\max(\text{SPS})}\right),$$

where SPS denotes the number of test samples processed per second using a fixed evaluation batch size of 8, and $\max(\text{SPS})$ is the highest mean SPS among the evaluated models for that benchmark. Higher values indicate a stronger performance–throughput trade-off. For the all-configuration analysis, mean Macro-F1 and mean SPS across the 60 configurations were used to calculate normalized efficiency. Normalization was performed separately within each benchmark to account for differences in dataset characteristics and inference workloads. For URBLIMP, we report the macro-average of category-wise accuracies.

[3] https://www.kaggle.com/datasets/akkefa/imdb-dataset-of-50k-movie-translated-urdu-reviews

5 Results

5.1 Pre-training Dynamics

During pre-training, perplexity was tracked on the validation set at each checkpoint (Figure 1). Across all three DunbaaBERT variants (32k, 52k, and 96k vocabularies), perplexity decreases rapidly during the early stages of training, followed by smooth and stable convergence. After the initial steep drop, convergence becomes notably flatter, with only marginal improvements after approximately 40k–50k updates, suggesting that the models approach saturation well before the end of training.

Notably, the perplexity trajectories of all three vocabulary configurations are almost indistinguishable. While minor fluctuations can be observed, these are transient and do not indicate instability. All models exhibit highly similar convergence behavior, pointing to stable optimization and good generalization.

A particularly interesting observation is that vocabulary size appears to have only negligible influence on perplexity. Despite substantial differences between 32k, 52k, and 96k token vocabularies, the validation curves nearly overlap throughout training, with final validation perplexities differing only marginally (4.59, 4.52, and 4.35, respectively). While the 96k vocabulary achieves the numerically lowest perplexity, the absolute differences remain small, suggesting that intrinsic language modeling behavior is governed primarily by the characteristics of the pre-training corpus rather than by vocabulary size itself. In other words, increasing vocabulary size beyond 32k yields only diminish-

Model	Asp. Agr.	DatObj N.	DatObj Pr.	Erg. VGDat.	Erg. PerfV.	Erg. ObjVAg.	Exp. Subj.	Honorific	N-Adj	Obl. SN.	Obl. Pron.	Obl. Adj.	Obl. Plur.	Obl. Verb	Part. Rel.	S-V Gender	S-V Num.	S-V Pers.	WordOrdVar	**AVG**
DunbaaBERT$_{32k}$	**100.0**	95.6	**99.2**	84.5	99.6	88.5	95.8	85.6	99.0	**95.0**	98.5	**98.0**	93.2	98.0	95.7	97.8	**98.5**	91.1	93.1	95.1
DunbaaBERT$_{52k}$	**100.0**	92.5	98.5	95.6	99.8	90.0	**98.3**	90.8	**99.5**	94.0	99.0	**98.0**	**96.1**	98.0	97.3	**99.3**	**98.5**	**99.0**	98.0	97.0
DunbaaBERT$_{96k}$	98.5	95.6	95.5	94.5	98.7	91.0	90.6	92.8	97.0	88.0	97.2	95.1	94.2	96.1	97.0	93.3	97.6	98.0	87.1	94.6
Urdu-RoBERTa$_{small}$	99.8	86.9	90.9	96.0	99.6	86.5	95.8	71.2	92.5	89.0	99.0	84.3	84.5	**99.0**	89.7	81.6	89.8	95.1	88.1	90.5
HPLT-BERT$_{ur}$	98.3	**98.8**	**99.2**	**98.4**	**99.9**	**97.0**	96.5	**94.8**	97.5	**95.0**	**99.2**	95.1	95.1	96.1	**98.0**	98.1	97.1	97.0	98.0	**97.3**
mBERT	98.8	65.0	53.0	85.1	95.7	63.0	18.5	58.2	86.5	77.0	93.7	74.5	78.6	95.1	82.4	73.8	88.3	55.4	91.1	75.5
mmBERT$_{small}$	99.3	88.1	96.2	93.3	98.4	82.0	73.8	65.4	87.0	91.0	93.3	95.1	84.5	96.1	89.7	85.4	91.3	92.1	**99.0**	89.5
mmBERT$_{base}$	98.0	90.0	97.7	94.5	99.6	89.0	89.4	71.2	94.0	92.0	97.6	95.1	85.4	94.1	92.0	91.4	91.7	95.7	97.0	92.4
XLM-R$_{base}$	99.5	93.1	94.7	96.9	99.3	72.5	92.8	61.4	88.5	89.0	99.0	91.2	89.3	94.1	79.4	84.6	88.3	90.2	98.0	89.6
XLM-R$_{large}$	99.3	95.0	97.7	90.0	99.8	92.0	95.3	79.7	93.5	92.0	98.4	92.2	88.3	97.1	92.0	93.6	96.1	98.7	98.0	94.1

Table 2: Detailed URBLIMP evaluation across 19 linguistic acceptability categories. Best results are shown in bold and second-best results are underlined.

ing returns in intrinsic modeling quality, motivating assessment of vocabulary choices mainly through downstream performance and computational efficiency rather than perplexity alone.

We further observe that the total number of completed epochs differs slightly across vocabulary configurations at a fixed budget of 100k update steps. The 32k, 52k, and 96k models reach approximately 147, 150, and 152 epochs, respectively. This variation is expected, as vocabulary size affects sequence packing efficiency and thus the number of effective training examples processed per update. However, despite these small differences in epoch counts, all models exhibit nearly identical convergence behavior and final perplexity values, reinforcing the observation that vocabulary size has only a minor impact on intrinsic pre-training dynamics in our setting.

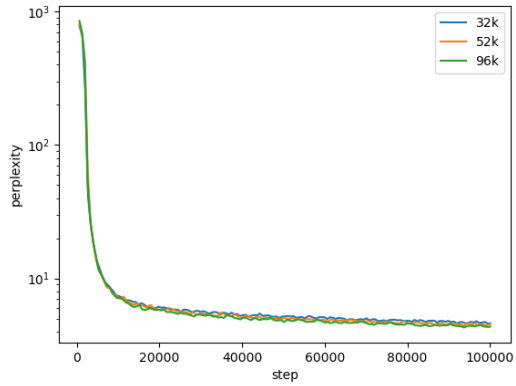


Figure 1: Validation perplexity across DunbaaBERT models with 32k, 52k, and 96k vocabularies. Nearly overlapping convergence curves and only marginal differences in final perplexity suggest limited impact of vocabulary size on intrinsic pre-training behavior.

5.2 Linguistic Acceptability

Table 2 reports detailed results on URBLIMP, evaluating fine-grained syntactic and morphosyntactic competence across 19 linguistic categories. Across all DunbaaBERT variants, we observe consistently strong performance, substantially outperforming multilingual baselines such as mBERT, mmBERT, and XLM-R. The strongest overall results are achieved by HPLT-BERT$_{\text{ur}}$. DunbaaBERT remains competitive with HPLT-BERT$_{\text{ur}}$ while using a controlled and comparatively compact pre-training setup, highlighting the effectiveness of our approach.

Focusing on the vocabulary ablation, the DunbaaBERT models exhibit a clear but nuanced pattern. DunbaaBERT$_{52k}$ achieves the best overall performance with an average score of 97.0, followed by DunbaaBERT$_{32k}$ (95.1) and DunbaaBERT$_{96k}$ (94.6). While all three models perform at a high level, the differences between them are more pronounced than those observed in pre-training perplexity, indicating that downstream linguistic competence is more sensitive to vocabulary design than intrinsic language modeling metrics.

Interestingly, the medium-sized vocabulary (52k) yields the strongest and most balanced results across categories, achieving top or near-top performance in multiple areas, including agreement, oblique marking, and subject–verb agreement. In contrast, the 32k model performs competitively but shows slightly weaker performance in more complex agreement and word order categories. The 96k model, despite achieving the lowest perplexity during pre-training, does not translate this advantage into improved downstream performance and instead exhibits small but consistent degradations across several categories.

Overall, under our controlled experimental setup, increasing the vocabulary size from 52k to 96k did not improve fine-grained grammatical performance. Because the DunbaaBERT variants share the same architecture, pre-training corpus, training budget, and optimization settings, these results isolate vocabulary size as the primary experimental difference. However, they should not be interpreted as establishing a universally optimal vocabulary size for Urdu.

5.3 Downstream Tasks

Table 3 summarizes the downstream results for news domain classification, offensive language detection, and sentiment classification. Variability across the complete hyperparameter search space is reported in Appendix D.3, Table 7, which summarizes 60 configurations per model and benchmark as mean $\pm$ standard deviation. Additional search-cost and efficiency analyses are provided in Figure 4. Overall, the DunbaaBERT variants achieve competitive task performance while consistently maintaining strong normalized efficiency across benchmarks. In contrast, larger multilingual models can achieve strong raw classification scores, but at substantially weaker efficiency trade-offs.

News Domain Classification On COUNT19, HPLT$_{\text{base}}$ achieves the highest Macro-F1 and accu-

Model	COUNT19			USADC			PSL–Kabaddi			IMDB Urdu		
	F1	Acc.	Norm. Eff.	F1	Acc.	Norm. Eff.	F1	Acc.	Norm. Eff.	F1	Acc.	Norm. Eff.
DunbaaBERT$_{32k}$	94.44	95.17	**0.944**	**94.08**	**94.15**	0.893	70.08	82.51	**0.701**	90.13	90.14	**0.897**
DunbaaBERT$_{52k}$	94.91	95.46	0.908	91.75	91.81	0.895	67.60	81.40	0.634	90.14	90.15	0.744
DunbaaBERT$_{96k}$	95.22	95.96	0.917	89.97	90.06	**0.900**	70.53	**82.78**	0.655	90.65	90.66	0.779
Urdu-RoBERTa$_{small}$	92.08	93.39	0.890	85.36	85.38	0.792	67.06	77.55	0.595	84.72	84.73	0.847
HPLT$_{base}$	**95.71**	**96.35**	0.585	93.51	93.57	0.764	**71.11**	81.54	0.487	89.69	89.70	0.553
mBERT	90.88	92.37	0.825	83.03	83.04	0.716	65.78	76.03	0.587	85.47	85.47	0.846
mmBERT$_{small}$	92.36	93.43	0.513	73.09	73.10	0.570	70.36	79.48	0.432	85.44	85.44	0.461
mmBERT$_{base}$	93.97	94.71	0.522	77.77	77.78	0.582	67.75	78.37	0.409	87.31	87.31	0.468
XLM-R$_{base}$	93.72	94.54	0.910	85.22	85.38	0.712	60.56	79.75	0.539	88.69	88.70	0.855
XLM-R$_{large}$	94.38	95.04	0.512	83.55	83.63	0.576	69.62	82.64	0.405	**91.15**	**91.15**	0.476

Table 3: Results across four Urdu downstream benchmarks. Norm. Eff. denotes normalized efficiency (see **??**). Best results are shown in bold and second-best results are underlined.

racy scores (95.71 and 96.35), followed closely by DunbaaBERT$_{96k}$ (95.22 and 95.96). However, the DunbaaBERT variants achieve substantially stronger efficiency trade-offs, with DunbaaBERT$_{32k}$ obtaining the highest normalized efficiency score of 0.944. The broader efficiency analysis in Appendix D.3 further shows that the DunbaaBERT variants maintain favorable throughput and training-time behavior across the full hyperparameter search space.

Offensive Language Detection On USADC, DunbaaBERT$_{32k}$ achieves the strongest overall performance with a Macro-F1 score of 94.08 and an accuracy of 94.15. The DunbaaBERT variants also dominate normalized efficiency, with DunbaaBERT$_{96k}$ achieving the highest score (0.900) and DunbaaBERT$_{52k}$ the second-highest score (0.895). HPLT$_{base}$ remains competitive with a Macro-F1 score of 93.51, whereas multilingual baselines such as mmBERT and XLM-R show noticeably weaker downstream effectiveness and efficiency.

Sentiment Classification For sentiment classification, the results differ across the two benchmarks. On PSL–Kabaddi, HPLT$_{base}$ achieves the highest Macro-F1 score (71.11), while DunbaaBERT$_{96k}$ obtains the highest accuracy (82.78). Interestingly, mmBERT$_{small}$ also performs competitively on this benchmark, reaching a Macro-F1 score of 70.36 despite comparatively weak normalized efficiency. On IMDB Urdu, XLM-R$_{large}$ achieves the highest Macro-F1 and accuracy scores (91.15), indicating that larger multilingual encoders may benefit from broader semantic coverage on large-scale document-level sentiment tasks. Nevertheless, DunbaaBERT$_{32k}$ achieves the strongest normalized efficiency score (0.897), substantially outperforming the larger multilingual baselines in terms of performance–efficiency trade-off. The search-cost-aware analysis in Figure 4 further illustrates that the DunbaaBERT variants consistently occupy favorable performance–efficiency regions relative to their overall search cost.

Across downstream tasks, the results indicate that increasing the DunbaaBERT vocabulary size from 32k or 52k to 96k does not consistently improve downstream effectiveness under otherwise identical experimental conditions. While DunbaaBERT$_{96k}$ often achieves slightly stronger raw classification scores, DunbaaBERT$_{32k}$ repeatedly provides the strongest overall efficiency profile. Moreover, although multilingual models such as XLM-R$_{large}$ and HPLT$_{base}$ can achieve highly competitive predictive performance, the Urdu-specific DunbaaBERT variants generally provide substantially stronger cost-efficiency trade-offs.

6 Discussion

The observed trends in UrBLiMP exhibit notable parallels to findings previously reported for Turkish in the SindBERT study (Schmitt and Schweter, 2026). In particular, SindBERT showed that larger parameter counts and scaling-oriented multilingual approaches do not necessarily translate into consistently superior linguistic competence for morphologically rich languages. Our results point toward similar dynamics for Urdu. Although the 96k DunbaaBERT configuration achieved the lowest pre-training perplexity, this advantage did not consistently transfer to downstream linguistic acceptability performance, where the medium-sized 52k configuration yielded the strongest overall results.

Motivated in part by earlier observations from SindBERT, we placed particular emphasis on corpus quality, diversity, and deduplication during corpus construction; instead of relying solely on raw crawl scale, we deliberately combined multiple heterogeneous Urdu sources and selected CulturaX

as a comparatively clean, pre-filtered, and deduplicated web corpus component. Furthermore, the NLLB-derived portion of the corpus underwent additional filtering and processing, reducing the overall corpus size from approximately 22.3GB before deduplication to roughly 17GB in the final training corpus. Taken together, these observations suggest that careful corpus curation, tokenizer construction, and training quality may contribute more strongly to downstream linguistic competence than vocabulary scaling alone.

This interpretation is further supported by the strong performance of HPLT-BERT$_{ur}$ and the medium-sized DunbaaBERT$_{52k}$ configuration, both of which balance vocabulary coverage with comparatively compact subword inventories. At the same time, the strong HPLT-BERT$_{ur}$ results raise additional questions regarding the interaction between corpus composition, pre-training objectives, and linguistic acceptability evaluation. In particular, we speculate that the HPLT corpus construction pipeline may ultimately provide training data that are somewhat better aligned with BLiMP-style grammatical phenomena than our own corpus mixture.

Another open question concerns the role of sentence-level pre-training objectives such as next sentence prediction (NSP). While RoBERTa-style training commonly omits NSP, several strong-performing BERT-family models in morphologically rich languages suggest that discourse-level objectives may still warrant further investigation, particularly for free-word-order and agreement-sensitive phenomena.

Beyond the intrinsic evaluation, the downstream results highlight the effectiveness of the DunbaaBERT models across practical Urdu NLP tasks. Across multiple Urdu classification benchmarks, the DunbaaBERT variants achieved highly competitive results against substantially larger multilingual baselines while consistently maintaining stronger normalized efficiency. In particular, DunbaaBERT$_{32k}$ repeatedly provided the strongest overall performance–efficiency trade-off within our experimental setting, indicating that vocabulary scaling alone did not guarantee improved downstream effectiveness in the evaluated models and tasks. Notably, on the USADC benchmark, the DunbaaBERT variants also outperform previously reported GPT-4-based Urdu offensive-language detection results discussed by Maab et al. (2026), despite relying on comparatively compact encoder-only architectures. Taken together, our findings indicate that carefully trained Urdu-specific encoder models can remain highly competitive for practical downstream NLP applications, particularly under realistic efficiency and deployment constraints.

More broadly, our findings highlight several promising directions for future Urdu language modeling. While the present work focuses on standard encoder-only architectures with a maximum sequence length of 512 tokens, recent work on long-context transformers suggests that extended-context training may provide additional benefits for discourse-sensitive and document-level phenomena. Architectures such as Longformer (Beltagy et al., 2020), Nyströmformer (Xiong et al., 2021), and recent efficient long-context encoder approaches, including ModernBERT (Warner et al., 2024), may therefore represent promising directions for Urdu representation learning, particularly given the prevalence of long-form news articles and document-centric web corpora.

Finally, the current Urdu evaluation landscape still lacks broad standardized downstream benchmark suites. While UrBLiMP provides valuable insight into grammatical acceptability and morphosyntactic competence, broader evaluation settings covering semantic understanding, reasoning, classification, retrieval, and sentence-pair tasks remain comparatively limited. Although GLUE-style Urdu benchmark collections have recently been proposed, we did not include them because the identified resource was available only as an anonymous preprint at the time of this study.[4]

7 Conclusion

We introduced DunbaaBERT, a family of Urdu RoBERTa-base models trained with different Byte-BPE vocabulary sizes and evaluated across intrinsic and downstream benchmarks. Overall, the results show that DunbaaBERT is competitive with strong Urdu and multilingual baselines, while the 32k variant often provides the strongest performance–efficiency trade-off. We release the DunbaaBERT models under the MIT license.

Limitations

A limitation of the present work is that the downstream evaluation focuses primarily on classification benchmarks. Based on our extensive survey of

[4] https://openreview.net/forum?id=ESTMRKI7WK

currently available Urdu resources, these tasks provided the most reliable and reproducible evaluation setting. In contrast, existing Urdu NER datasets often exhibit substantial limitations, including incomplete annotations, limited documentation, or unusually saturated performance levels reported in prior work (Kanwal et al., 2019; Zafar et al., 2025), which complicates meaningful comparison and analysis. Furthermore, although GLUE-style Urdu benchmark collections have recently emerged (Anonymous, 2026), we did not include them in the present work since the benchmark currently exists only as an anonymous pre-print. Nevertheless, NER and broader benchmark standardization remain important directions for future work, particularly since vocabulary size and tokenization granularity may have stronger effects on sequence labeling tasks than on document-level classification.

In addition, the pre-training corpus was derived primarily from web-based sources and therefore may not fully capture regional, dialectal, or domain-specific variation in Urdu. Like other web-derived corpora, it may also contain social and linguistic biases present in online text.

Finally, although our controlled comparison varies vocabulary size while holding the pre-training corpus, architecture, training budget, and optimization settings constant, we do not report tokenizer-level diagnostics such as subwords per word, vocabulary utilization, or morphology-sensitive segmentation quality. Consequently, our results characterize empirical performance differences among the three vocabulary sizes but do not establish the tokenization mechanisms underlying these differences. Future work should combine such diagnostics with qualitative error analyses to clarify vocabulary- and tokenization-related performance patterns.

Ethical Considerations

Like other large-scale language models trained on web-derived corpora, DunbaaBERT may inherit social and linguistic biases present in online text, which can affect downstream applications. In addition, although the training corpus was filtered and deduplicated, large-scale web data may still contain noisy or potentially sensitive content. Finally, pre-training transformer models requires substantial computational resources and therefore contributes to energy consumption and environmental impact.

Acknowledgments

The authors dedicate this work in remembrance of the faith of Abraham and in gratitude for the guidance, grace, and mercy that accompanied them throughout this journey. Further, the authors gratefully acknowledge the scientific support and resources of the AI service infrastructure LRZ AI Systems provided by the Leibniz Supercomputing Centre (LRZ) of the Bavarian Academy of Sciences and Humanities (BAdW), funded by Bayerisches Staatsministerium für Wissenschaft und Kunst (StMWK), and Research and Development Center for Large Language Models (LLMC), National Institute of Informatics, Tokyo.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Sarfraz Ahmad, Hasan Iqbal, Momina Ahsan, Numaan Naeem, Muhammad Ahsan Riaz Khan, Arham Riaz, Muhammad Arslan Manzoor, Yuxia Wang, and Preslav Nakov. 2025. UrduFactCheck: An agentic fact-checking framework for Urdu with evidence boosting and benchmarking. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 22788–22802, Suzhou, China. Association for Computational Linguistics.

Rabee Adel Al-Qasem. 2026. U-rocx: An xlstm based approach to ai-generated urdu text detection. In *Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script*, pages 443–447.

Mehreen Alam and Sibt Ul Hussain. 2022. Roman-urdu-parl: Roman-urdu and urdu parallel corpus for urdu language understanding. *Transactions on Asian and Low-Resource Language Information Processing*, 21(1):1–20.

Maaz Amjad, Alisa Zhila, Grigori Sidorov, Andrey Labunets, Sabur Butta, Hamza Imam Amjad, Oxana Vitman, and Alexander Gelbukh. 2022. Overview of abusive and threatening language detection in urdu at fire 2021. *arXiv preprint arXiv:2207.06710*.

saeed A. Anabtawi. 2026. A stylometric and statistical pipeline for Urdu AI-generated text classification. In *Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script*, pages 472–475, Rabat, Morocco. Association for Computational Linguistics.

Anonymous. 2026. Urdu-GLUE: A comprehensive benchmark and dynamic prompt-based fine-tuning for urdu language understanding. In *Submitted to ACL Rolling Review - January 2026*. Under review.

Zahra Ansari, Shaukat Ali, and Fatima Khan. 2020. Use of roman script for writing urdu language. *International Journal of Linguistics and Culture*, 1(2):165–178.

Samee Arif, Abdul Hameed Azeemi, Agha Ali Raza, and Awais Athar. 2024a. Generalists vs. specialists: Evaluating large language models for urdu. *arXiv preprint arXiv:2407.04459*.

Samee Arif, Abdul Hameed Azeemi, Agha Ali Raza, and Awais Athar. 2024b. 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 Umair Arshad, Raza Ali, Mirza Omer Beg, and Waseem Shahzad. 2023. Uhated: hate speech detection in urdu language using transfer learning. *Language Resources and Evaluation*, 57(2):713–732.

Muhammad Rehan Ashraf, Muzammal Hussain, M Arfan Jaffar, Waheed Yousuf Ramay, and Muhammad Faheem. 2024. Revolutionizing urdu sentiment analysis: Harnessing the power of xlm-r and gpt-2. *IEEE Access*.

Muhammad Rehan Ashraf, Yasmeen Jana, Qasim Umer, M. Arfan Jaffar, Sungwook Chung, and Waheed Yousuf Ramay. 2023. BERT-based sentiment analysis for low-resourced languages: A case study of Urdu language. *IEEE Access*, 11:110245–110259.

Elena Bashir. 2011. Urdu and linguistics: A fraught but evolving relationship.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *Preprint*, arXiv:2004.05150.

Muhammad Bilal, Atif Khan, Salman Jan, Shahrulniza Musa, and Shaukat Ali. 2023. Roman urdu hate speech detection using transformer-based model for cyber security applications. *Sensors*, 23(8):3909.

Damian Blasi, Antonios Anastasopoulos, and Graham Neubig. 2021. Systematic inequalities in language technology performance across the world's languages. *arXiv preprint arXiv:2110.06733*.

Nicolas Boizard, Hippolyte Gisserot-Boukhlef, Duarte M. Alves, André Martins, Ayoub Hammal, Caio Corro, Céline Hudelot, Emmanuel Malherbe, Etienne Malaboeuf, Fanny Jourdan, Gabriel Hautreux, João Alves, Kevin El-Haddad, Manuel Faysse, Maxime Peyrard, Nuno M. Guerreiro, Patrick Fernandes, Ricardo Rei, and Pierre Colombo. 2025. Eurobert: Scaling multilingual encoders for european languages. *Preprint*, arXiv:2503.05500.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Ona de Gibert, Graeme Nail, Nikolay Arefyev, Marta Bañón, Jelmer van der Linde, Shaoxiong Ji, Jaume Zaragoza-Bernabeu, Mikko Aulamo, Gema Ramírez-Sánchez, Andrey Kutuzov, Sampo Pyysalo, Stephan Oepen, and Jörg Tiedemann. 2024. A new massive multilingual dataset for high-performance language technologies. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1116–1128, Torino, Italia. ELRA and ICCL.

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

Pummy Dhiman, Amandeep Kaur, Deepali Gupta, Sapna Juneja, Ali Nauman, and Ghulam Muhammad. 2024. Gbert: A hybrid deep learning model based on gpt-bert for fake news detection. *Heliyon*, 10(16).

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *Preprint*, arXiv:2010.11125.

Maria Isabel Maldonado Garcia. 2011. The urdu language reforms. *Studies*, 26:97.

Syed Ali Hamza, Bilal Tahir, and Muhammad Amir Mehmood. 2019. Domain identification of urdu news text. In *2019 22nd International Multitopic Conference (INMIC)*, pages 1–7.

Muhammad Ehtisham Hassan, Iffat Maab, Masroor Hussain, Usman Habib, and Yutaka Matsuo. 2024. Polarity classification of low resource roman urdu and movie reviews sentiments using machine learning-based ensemble approaches. *IEEE Open Journal of the Computer Society*, 5:599–611.

Ammarah Irum and M Ali Tahir. 2025. Document-level sentiment analysis of urdu text using deep learning techniques. *arXiv preprint arXiv:2501.17175*.

Safia Kanwal, Kamran Malik, Khurram Shahzad, Faisal Aslam, and Zubair Nawaz. 2019. Urdu named entity recognition: Corpus generation and deep learning applications. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(1).

Usama Khalid, Mirza Omer Beg, and Muhammad Umair Arshad. 2021. Rubert: A bilingual roman urdu bert using cross lingual transfer learning. *Preprint*, arXiv:2102.11278.

Hasan Faraz Khan, Noor Fatima, and Irfan Ahmad. 2026. Enhancing Urdu sentiment classification through instruction-tuned LLMs and cross-lingual transfer. In *Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script*, pages 198–207, Rabat, Morocco. Association for Computational Linguistics.

Lal Khan, Ammar Amjad, Noman Ashraf, and Hsien-Tsung Chang. 2022. Multi-class sentiment analysis of urdu text using multilingual bert. *Scientific Reports*, 12(1):5436.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Iffat Maab, Usman Haider, and Junichi Yamagishi. 2026. Prompt-driven detection of offensive Urdu language using large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4302–4327, Rabat, Morocco. Association for Computational Linguistics.

Zoya Maqsood. 2023. Weakly supervised learning for aspect based sentiment analysis of Urdu tweets. In *Proceedings of the 8th Student Research Workshop associated with the International Conference Recent Advances in Natural Language Processing*, pages 78–86, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. mmbert: A modern multilingual encoder with annealed language learning. *Preprint*, arXiv:2509.06888.

AI Meta. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. *Meta AI*.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *Preprint*, arXiv:2309.09400.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Tariq Rahman. 1997. The urdu—english controversy in pakistan. *Modern Asian Studies*, 31(1):177–207.

Muhammad Owais Raza, Aqsa Umar, and Mehrub Awan. 2025. Slur and emoji aware models for hate and sentiment detection in roman urdu transgender discourse. In *Proceedings of the First Workshop on Advancing NLP for Low-Resource Languages*, pages 131–139.

Hafiz Hassaan Saeed, Muhammad Haseeb Ashraf, Faisal Kamiran, Asim Karim, and Toon Calders. 2021. Roman urdu toxic comment classification. *Language Resources and Evaluation*, pages 1–26.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

Raphael Scheible-Schmitt, Henry He, and Armando B. Mendes. 2025. PortBERT: Navigating the depths of Portuguese language models. In *Proceedings of the Workshop on Beyond English: Natural Language Processing for all Languages in an Era of Large Language Models*, pages 59–71, Varna, Bulgaria. INCOMA Ltd., Shoumen, BULGARIA.

Raphael Schmitt. 2026. From raw text to fairseq roberta: A modular snakemake-based framework enabling language-specific bpe tokenization. *Software Impacts*, 27:100824.

Raphael Schmitt and Stefan Schweter. 2026. SindBERT, the sailor: Charting the seas of Turkish NLP. In *Proceedings of the Second Workshop Natural Language Processing for Turkic Languages (SIGTURK 2026)*, pages 1–13, Rabat, Morocco. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2020. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *Preprint*, arXiv:1911.04944.

Muhammad Haroon Shakeel and Asim Karim. 2020. Adapting deep learning for sentiment classification of code-switched informal short text. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, SAC '20, page 903–906, New York, NY, USA. Association for Computing Machinery.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Cagri Toraman, Eyup Halit Yilmaz, Furkan Şahiṅuç, and Oguzhan Ozcelik. 2023. Impact of tokenization on language models: An analysis for turkish. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(4):1–21.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, and Marie-Anne Lachaux. 2023. Timothé e lacroix, baptiste roziè re, naman goyal, eric hambro, faisal azhar, auré lien rodriguez, armand joulin, edouard grave, and guillaume lample. 2023 a. llama: Open and efficient foundation language models. corr, vol. abs/2302.13971 (2023). *LLaMA: Open and Efficient Foundation Language Models. CoRR, Vol. abs/2302.13971*.

Kurban Ubul, Gulzira Tursun, Alimjan Aysa, Donato Impedovo, Giuseppe Pirlo, and Tuergen Yibulayin. 2017. Script identification of multi-script documents: a survey. *IEEE access*, 5:6546–6559.

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

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. 2021. Nyströmformer: A nyström-based algorithm for approximating self-attention. *Preprint*, arXiv:2102.03902.

Adeel Zafar, Sohail Ashraf, and Slawomir Nowaczyk. 2025. From courtroom to corpora: Building a name entity corpus for Urdu legal texts. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing - Natural Language Processing in the Generative AI Era*, pages 1396–1405, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

A Technical Specifications

A.1 Computational Setup.

All experiments were conducted on one GPU compute node. GPU compute node is equipped with NVIDIA A100 GPU with 40GB memory. The models were fine-tuned using PyTorch and HuggingFace Transformers, with mixed-precision training enabled where applicable.

A.2 Implementation Specifics

For all downstream classification benchmarks, we fine-tuned each model under a unified hyperparameter-search setting. We searched over learning rates $\{5e{-}6, 7e{-}6, 1e{-}5, 2e{-}5, 5e{-}5\}$ and training batch sizes $\{16, 32, 48, 64\}$ using three random seeds $\{5, 42, 777\}$. Models were trained for up to 30 epochs with early stopping patience of 3, and the best configuration was selected according to validation macro-F1. For final reporting, we evaluated the selected configuration on the held-out test set and report mean $\pm$ standard deviation across seeds. To ensure comparable efficiency measurements, evaluation and inference were performed with a fixed batch size of 8. In addition to accuracy and macro-F1, we also record computational efficiency metrics, including wall-clock training runtime, and inference throughput in samples per second.

For URBLIMP, we compute accuracy for each category and report the macro-average across all categories as the overall URBLIMP score. This evaluation captures the models' ability to distinguish fine-grained grammatical patterns beyond surface-level classification performance. Furthermore, for all other benchmarks, we use an 80/10/10 train/validation/test split and select hyperparameters using validation macro-F1. For COUNT19, a seven-way Urdu news-domain classification dataset, the split is stratified over news categories. For USADC, we use stratified splits over the binary labels, *Normal* and *Abusive*. For the PSL–Kabaddi sentiment benchmark, we stratify the split by sentiment polarity to preserve the distribution of positive, negative, and neutral examples. For the Urdu IMDB dataset, which contains 50,000 balanced positive and negative movie reviews, we likewise use an 80/10/10 split, with the validation set reserved for hyperparameter selection and the held-out test set used only for final reporting. Across all datasets, the reported test results are obtained from the best learning-rate and batch-size configuration selected according to mean validation macro-F1 across the three random seeds. The selected hyperparameters for each model and downstream benchmark are reported in Table 4.

B Qualitative Analysis of Heuristic Penalties

To better understand the behavior of the proposed heuristic filtering approach, Table 5 presents the five highest-ranked and five lowest-ranked candidates according to the final scoring function. These examples provide a qualitative sanity check illustrating both which types of textual fragments are preserved and which receive the strongest penalties during corpus construction.

The lowest-ranked candidates are dominated by typical web-crawling artifacts, including truncated navigation snippets, "Read more" fragments, timestamps, category labels, newsroom metadata, and malformed or noisy Unicode remnants. Interestingly, several of these examples still achieve moderately high semantic similarity scores despite being linguistically uninformative. This observation suggests that embedding-based similarity alone is insufficient for reliable corpus filtering in large-scale crawl-based settings.

In contrast, the highest-ranked candidates largely consist of coherent and informative Urdu text fragments, including encyclopedic descriptions and geographically grounded narrative content. Although minor traces of noise or mixed-language artifacts remain present in some examples, the overall linguistic quality is substantially higher than in the heavily penalized candidates.

Instead, the results indicate that the additional heuristic penalty component successfully suppresses non-linguistic boilerplate and template artifacts that would otherwise remain in the final corpus. The examples further illustrate the importance of combining semantic scoring with lightweight rule-based filtering during corpus curation for morphologically rich low-resource languages such as Urdu.

C Model Properties

Table 6 summarizes the vocabulary sizes and parameter counts of the Urdu and multilingual models considered in our evaluation. Urdu-RoBERTa$_{\text{small}}$ (126M) and HPLT$_{\text{base}}$ (124M) represent strong monolingual Urdu baselines, while mmBERT$_{\text{small}}$ (140M) and mmBERT$_{\text{base}}$ (307M) provide multi-

Model	COUNT19		USADC		PSL		IMDB	
	BS	LR	BS	LR	BS	LR	BS	LR
DunbaaBERT$_{96k}$	48	1e-5	32	1e-5	16	2e-5	48	2e-5
DunbaaBERT$_{52k}$	48	1e-5	48	1e-5	64	5e-5	16	2e-5
DunbaaBERT$_{32k}$	32	5e-6	16	2e-5	32	5e-5	64	5e-5
HPLT-BERT$_{ur}$	64	5e-6	16	5e-5	64	5e-5	16	1e-5
XLM-R$_{base}$	32	7e-6	16	5e-5	48	5e-5	48	2e-5
mmBERT$_{base}$	32	2e-5	32	5e-5	64	5e-5	32	2e-5
mBERT$_{base}$	16	7e-6	32	5e-5	16	2e-5	48	7e-6
mmBERT$_{small}$	64	5e-5	32	2e-5	48	2e-5	64	5e-5
Urdu-RoBERTa$_{small}$	48	5e-5	32	5e-5	48	2e-5	32	5e-6
XLM-R$_{large}$	16	5e-6	48	5e-5	48	2e-5	64	5e-6

Table 4: Hyperparameters of the best downstream task models for each task and pre-trained model. BS denotes batch size and LR denotes learning rate. For each dataset, the reported configuration is selected using the highest mean validation macro-F1 across seed runs.

lingual comparison points specifically designed for massively multilingual settings.

Our DunbaaBERT variants systematically investigate the impact of vocabulary size on model scaling and downstream performance. DunbaaBERT$_{32k}$ contains 111M parameters and uses a compact 32k-token vocabulary, while DunbaaBERT$_{52kbase}$ (126M) closely matches the parameter scale of Roberta-Urdu and HPLT. The larger DunbaaBERT$_{96kbase}$ increases the vocabulary size to 96k tokens, resulting in a substantially larger parameter count of 160M.

For multilingual points of reference, mBERT contains 178M parameters with a WordPiece vocabulary of approximately 120k tokens, while XLM-R$_{base}$ and XLM-R$_{large}$ contain 278M and 560M parameters, respectively, using 250k-token SentencePiece vocabularies. All values were extracted using Huggingface's `transformers` library.

Model	Vocab Size	#Params
DunbaaBERT$_{32k}$	32009	110625024
HPLT-BERT$_{ur}$	32768	124361472
Urdu-RoBERTa$_{small}$	52000	125978112
DunbaaBERT$_{52k}$	52009	125985024
mmBERT$_{small}$	256000	140493696
DunbaaBERT$_{96k}$	96009	159777024
mBERT	119547	177853440
XLM-R$_{base}$	250002	278043648
mmBERT$_{base}$	256000	306939648
XLM-R$_{large}$	250002	559890432

Table 6: Vocabulary size and total parameter count for Urdu transformer-based models. Values were extracted using Huggingface's `transformers` library.

D Efficiency Performance

D.1 Predictive Performance and Inference Throughput

Figure 2 shows the relationship between predictive performance and inference throughput across the evaluated Urdu benchmarks. Unlike reporting macro-F1 alone, this analysis highlights whether models maintain strong performance while also supporting faster inference. The upper-right region is the most desirable, indicating both high test macro-F1 and high samples-per-second throughput. Across tasks, the DunbaaBERT variants frequently occupy this favorable region, suggesting that the proposed Urdu-specific models provide not only competitive performance across all tasks but also efficient inference behavior compared with multilingual and other Urdu baselines.

D.2 Downstream Training and Inference Efficiency

Figure 3 analyzes the efficiency–performance trade-off by combining downstream predictive quality and inference throughput into a single efficiency factor, defined as macro-F1 $\times$ inference samples per second. The plot complements the numerical results by showing how models compare not only in terms of final task performance, but also with respect to the downstream fine-tuning cost. All points are based on the best hyperparameter configurations selected through validation performance under the same HPO setting, with early stopping using patience 3. A model is preferable when it achieves a higher efficiency factor with lower training time, corresponding to the upper-left region of the plot. The DunbaaBERT variants consistently appear in this favorable region across

Final	Sim.	Pen.	Text
			Top 5 Candidates
0.9763	0.9763	0.0	مربع کلومیٹر کے 745.52 ستکوسیا صحابہ مجسمہ سبز کی ایک رگڑ ہے جو انجرو، نیرغھ اور فلوانی کے اضلاع میں ایک ادارے کی سطح پر پھیلا ہوا ہے۔
0.9681	0.9681	0.0	ڈیکٹر اراضی پر 650 کی اونچائی پر 2.000 چامباشی اسکی سنٹر میں جو سہولیات اوردو میٹروپولیٹن بلدیہ نے کلومیٹر دور ہے، سردیوں کے موسم کے لئے تیار ہو رہی ہے۔ 52 واقع ہے اور شہر کے مرکز سے
0.9672	0.9672	0.0	کلومیٹر مشرق میں واقع، اکسو ضلع کی سرحدوں کے اندر واقع ہے، جو ایک بار 18 انٹیلیا سے (Perge) پیرو پامفیلیا ریجن کا دارالحکومت ہے۔
0.9669	0.9669	0.0	کوکیمبو ایک بندرگاہی شہر، کیمون اور صوبہ الکوئی کا دارالحکومت ہے، جو پین امریکی شاہراہ پر چلی کے کوکیمبو ریجن میں واقع ہے۔
0.9664	0.9664	0.0	میں واقع ہے۔ Tavastia Proper فورسا فن لینڈ کا ایک بلدیہ اور قصبہ جو
			Worst 5 Candidates
-1.3478	0.5522	1.9	خیر پھر ... Read more ...
-1.3142	0.5858	1.9	فائل فوٹو Read More
-1.2865	0.6135	1.9	سنیے Read more
-1.2418	0.6582	1.9	1 day ago ویب ڈیسک
-1.2332	0.5668	1.8	& آؤمہ لیہآلیہ & 6 days ago

Table 5: Qualitative comparison between the five highest-ranked candidates (top) and five lowest-ranked candidates (bottom) according to the proposed heuristic scoring function. The columns report the final heuristic score (Final), semantic similarity score (Sim.), penalty value (Pen.), and candidate sentence (Text).

multiple benchmarks, indicating that they converge efficiently during downstream training while maintaining strong inference efficiency.

D.3 Performance–Efficiency Across all 60 Configurations

To assess whether the models are consistently efficient beyond a single best run, Table 7 reports average results over 60 hyperparameter configurations for each model and benchmark, corresponding to 5 learning rates $\times$ 4 batch sizes $\times$ 3 random seeds. This analysis combines predictive quality, inference speed, and downstream training time under the same early-stopping setting. Overall, the DunbaaBERT variants show strong performance–efficiency trade-offs across benchmarks, often achieving the highest or second-highest normalized efficiency while maintaining competitive Macro-F1 and relatively low fine-tuning time. This suggests that the Urdu-specific DunbaaBERT models are not only effective in terms of task performance, but also efficient across a broad range of training configurations.

D.4 Search-Cost–Aware Performance and Efficiency

Figure 4 further analyzes the relationship between total hyperparameter search cost and model effectiveness. Each point represents one model, where the x-axis reports the total training time required to evaluate the full search space across the four benchmarks, corresponding to 60 configurations per benchmark and 240 configurations per model. The left column reports scores from the HPO-selected setting, while the right column reports the corresponding all-configuration summaries.

Overall, the DunbaaBERT variants provide a strong search-cost–performance trade-off. In the HPO-selected setting, DunbaaBERT$_{32k}$ achieves the most favorable balance, combining high average Macro-F1, strong inference throughput, and the highest normalized efficiency while requiring relatively low total search cost. DunbaaBERT$_{96k}$ also performs competitively, particularly in Macro-F1 and normalized efficiency, but with slightly higher search cost. In contrast, HPLT-BERT$_{ur}$ obtains strong HPO-selected Macro-F1 but requires sub-

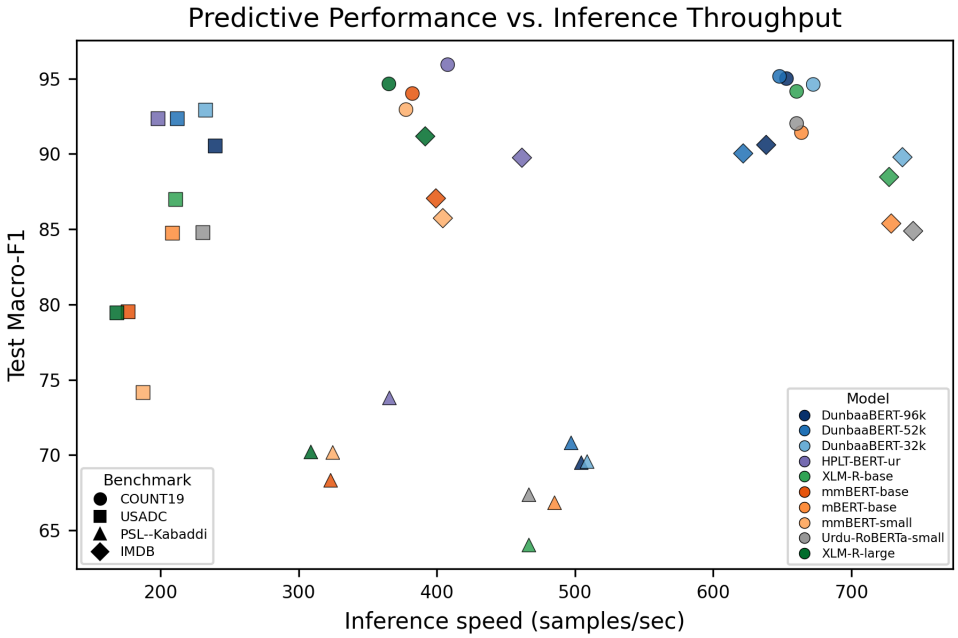


Figure 2: Predictive performance (test Macro-F1) versus inference throughput (samples per second) across downstream Urdu benchmarks. For each model and benchmark, results are based on the selected HPO configuration over learning rates and batch sizes, evaluated across three random seeds with early stopping patience 3. Marker shapes denote benchmark datasets, and colors denote model families.

stantially larger search time, weakening its overall cost-effectiveness.

The all-configuration results show a slightly different trend. DunbaaBERT$_{52k}$ appears particularly robust when averaging over the full configuration space, achieving strong Macro-F1 and normalized efficiency at relatively low search cost. This suggests that while DunbaaBERT$_{32k}$ is highly effective after HPO selection, DunbaaBERT$_{52k}$ may be more stable across broader hyperparameter settings. Larger multilingual models such as XLM-R$_{large}$ and HPLT-BERT$_{ur}$ generally require higher search cost, and their efficiency gains are less consistent. These results indicate that Urdu-specific pretraining can improve not only downstream accuracy but also the practical cost-efficiency of model selection.

E Computational Cost of Benchmarking

In addition to predictive performance and inference efficiency, we report the total downstream fine-tuning cost required for the benchmark suite. Table 8 summarizes the cumulative wall-clock training time across all evaluated configurations. For each benchmark, this includes 600 unique fine-tuning runs, corresponding to 10 models, 5 learning rates, 4 batch sizes, and 3 random seeds. Overall, the full evaluation required 2400 fine-tuning runs and 254 hours and 38 minutes of cumulative training time. This highlights the computational cost of systematic benchmarking and motivates reporting efficiency-oriented metrics alongside task performance.

Benchmark	Configs	Computation Time	Days
COUNT19	600	55:15	2.30
USADC	600	8:26	0.35
PSL–Kabaddi	600	15:15	0.64
IMDB Urdu	600	175:42	7.32
Total	**2400**	**254:38**	**10.61**

Table 8: Total downstream fine-tuning computation time across benchmarks. For each benchmark, computation time is summed over 600 unique fine-tuning runs, corresponding to 10 models $\times$ 5 learning rates $\times$ 4 batch sizes $\times$ 3 random seeds. Thus, each individual model contributes 60 configurations per benchmark. Computation Time is reported in hours:minutes and Days gives the equivalent duration in days.

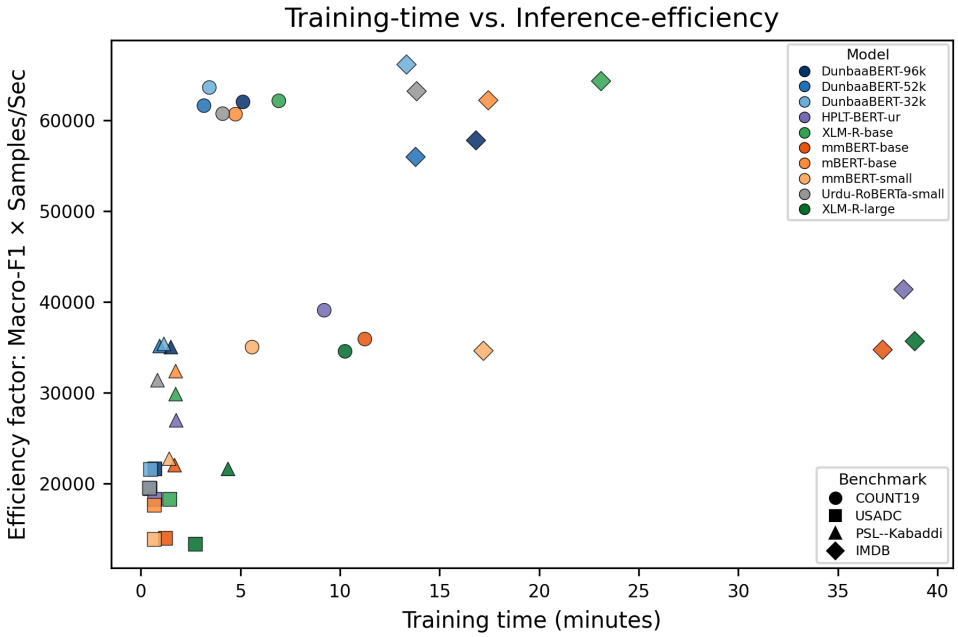
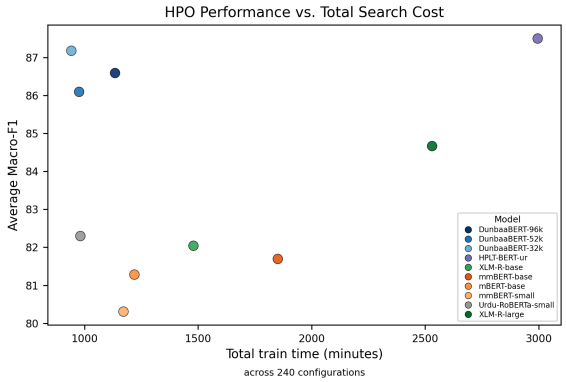


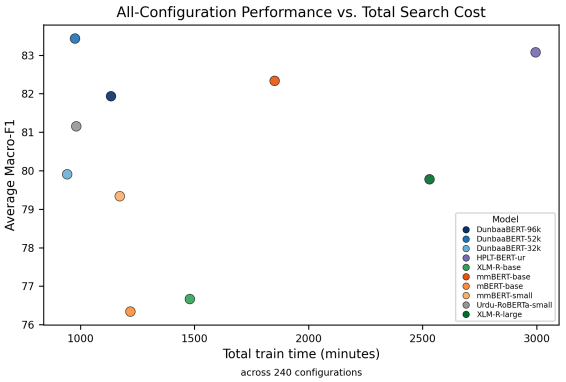
Figure 3: Training-time versus inference-efficiency trade-off across downstream Urdu benchmarks. The x-axis shows average fine-tuning time, and the y-axis shows the efficiency factor, defined as test Macro-F1 $\times$ inference samples per second. Results are based on the best HPO configuration across learning rates and batch sizes, averaged over three seeds with early stopping patience 3. Marker shapes indicate benchmarks, and colors indicate model families.

Benchmark	Model	Macro-F1	SPS	Train Time (min)	Norm. Eff.
COUNT19	DunbaaBERT$_{96k}$	$94.90_{\pm 0.50}$	$645.34_{\pm 14.48}$	$3.88_{\pm 1.37}$	0.916
	DunbaaBERT$_{52k}$	$94.86_{\pm 0.48}$	$654.38_{\pm 9.73}$	$\mathbf{3.18}_{\pm 0.85}$	$\underline{0.929}$
	DunbaaBERT$_{32k}$	$94.44_{\pm 0.58}$	$\mathbf{668.26}_{\pm 10.59}$	$3.33_{\pm 0.79}$	**0.944**
	HPLT-BERT$_{ur}$	$\mathbf{95.67}_{\pm 0.42}$	$413.68_{\pm 8.78}$	$8.80_{\pm 2.43}$	0.592
	XLM-R$_{base}$	$93.18_{\pm 0.86}$	$656.25_{\pm 10.61}$	$5.72_{\pm 1.97}$	0.915
	mmBERT$_{base}$	$93.12_{\pm 1.02}$	$382.11_{\pm 6.38}$	$7.17_{\pm 2.35}$	0.532
	mBERT$_{base}$	$90.57_{\pm 0.86}$	$635.04_{\pm 24.88}$	$4.65_{\pm 1.35}$	0.861
	mmBERT$_{small}$	$92.54_{\pm 0.76}$	$382.23_{\pm 5.70}$	$4.99_{\pm 1.58}$	0.529
	Urdu-RoBERTa$_{small}$	$91.74_{\pm 0.54}$	$\underline{663.41}_{\pm 8.70}$	$\underline{3.31}_{\pm 0.94}$	0.911
	XLM-R$_{large}$	$93.92_{\pm 1.12}$	$365.58_{\pm 4.26}$	$10.25_{\pm 3.69}$	0.514
USADC	DunbaaBERT$_{96k}$	$80.11_{\pm 20.21}$	$\mathbf{236.71}_{\pm 19.24}$	$0.54_{\pm 0.24}$	0.801
	DunbaaBERT$_{52k}$	$\mathbf{86.23}_{\pm 14.70}$	$\underline{235.66}_{\pm 25.42}$	$\underline{0.45}_{\pm 0.16}$	**0.858**
	DunbaaBERT$_{32k}$	$72.37_{\pm 25.03}$	$230.43_{\pm 10.86}$	$\mathbf{0.36}_{\pm 0.15}$	0.704
	HPLT-BERT$_{ur}$	$84.17_{\pm 12.04}$	$197.72_{\pm 9.04}$	$0.65_{\pm 0.24}$	0.703
	XLM-R$_{base}$	$68.13_{\pm 19.34}$	$223.20_{\pm 18.77}$	$1.10_{\pm 0.52}$	0.642
	mmBERT$_{base}$	$80.18_{\pm 4.35}$	$178.84_{\pm 7.26}$	$1.27_{\pm 0.30}$	0.606
	mBERT$_{base}$	$71.42_{\pm 12.32}$	$210.24_{\pm 11.51}$	$0.69_{\pm 0.22}$	0.634
	mmBERT$_{small}$	$71.30_{\pm 7.99}$	$184.14_{\pm 10.21}$	$0.70_{\pm 0.21}$	0.555
	Urdu-RoBERTa$_{small}$	$\underline{85.60}_{\pm 3.14}$	$224.36_{\pm 9.35}$	$0.54_{\pm 0.15}$	$\underline{0.811}$
	XLM-R$_{large}$	$71.32_{\pm 13.32}$	$165.80_{\pm 8.45}$	$2.14_{\pm 0.73}$	0.500
PSL–Kabaddi	DunbaaBERT$_{96k}$	$62.54_{\pm 5.44}$	$\underline{498.79}_{\pm 16.09}$	$0.97_{\pm 0.43}$	$\underline{0.593}$
	DunbaaBERT$_{52k}$	$62.61_{\pm 5.80}$	$498.10_{\pm 10.85}$	$\underline{0.88}_{\pm 0.38}$	0.592
	DunbaaBERT$_{32k}$	$63.09_{\pm 5.85}$	$\mathbf{526.39}_{\pm 11.10}$	$\mathbf{0.83}_{\pm 0.39}$	**0.631**
	HPLT-BER$_{ur}$	$63.00_{\pm 7.23}$	$364.45_{\pm 6.36}$	$1.33_{\pm 0.50}$	0.436
	XLM-R$_{base}$	$57.01_{\pm 3.89}$	$474.34_{\pm 7.64}$	$1.87_{\pm 0.66}$	0.514
	mmBERT$_{base}$	$\mathbf{69.30}_{\pm 3.90}$	$320.04_{\pm 7.02}$	$1.86_{\pm 0.48}$	0.421
	mBERT$_{base}$	$58.34_{\pm 7.44}$	$478.44_{\pm 18.21}$	$1.48_{\pm 0.60}$	0.530
	mmBERT$_{small}$	$\underline{68.44}_{\pm 3.45}$	$325.15_{\pm 4.50}$	$1.40_{\pm 0.40}$	0.423
	Urdu-RoBERTa$_{small}$	$62.82_{\pm 7.73}$	$477.33_{\pm 17.86}$	$1.03_{\pm 0.43}$	0.570
	XLM-R$_{large}$	$63.38_{\pm 9.06}$	$312.24_{\pm 7.13}$	$3.60_{\pm 1.42}$	0.376
IMDB Urdu	DunbaaBERT$_{96k}$	$\underline{90.19}_{\pm 0.38}$	$633.91_{\pm 19.91}$	$13.51_{\pm 5.74}$	0.771
	DunbaaBERT$_{52k}$	$90.05_{\pm 0.31}$	$672.41_{\pm 63.26}$	$11.76_{\pm 3.59}$	0.817
	DunbaaBERT$_{32k}$	$89.76_{\pm 0.29}$	$735.47_{\pm 10.79}$	$\mathbf{11.18}_{\pm 3.69}$	**0.890**
	HPLT-BERT$_{ur}$	$89.47_{\pm 0.29}$	$456.80_{\pm 14.47}$	$39.14_{\pm 12.06}$	0.551
	XLM-R$_{base}$	$88.35_{\pm 0.35}$	$722.30_{\pm 11.15}$	$15.97_{\pm 5.52}$	$\underline{0.861}$
	mmBERT$_{base}$	$86.74_{\pm 0.41}$	$404.76_{\pm 5.07}$	$20.55_{\pm 6.12}$	0.474
	mBERT$_{base}$	$85.05_{\pm 0.70}$	$\underline{737.58}_{\pm 8.61}$	$13.49_{\pm 5.04}$	0.846
	mmBERT$_{small}$	$85.08_{\pm 0.58}$	$404.45_{\pm 4.30}$	$12.44_{\pm 4.15}$	0.464
	Urdu-RoBERTa$_{small}$	$84.47_{\pm 0.55}$	$\mathbf{741.35}_{\pm 8.07}$	$\underline{11.48}_{\pm 3.34}$	0.845
	XLM-R$_{large}$	$\mathbf{90.52}_{\pm 1.02}$	$391.52_{\pm 4.64}$	$26.18_{\pm 9.45}$	0.478

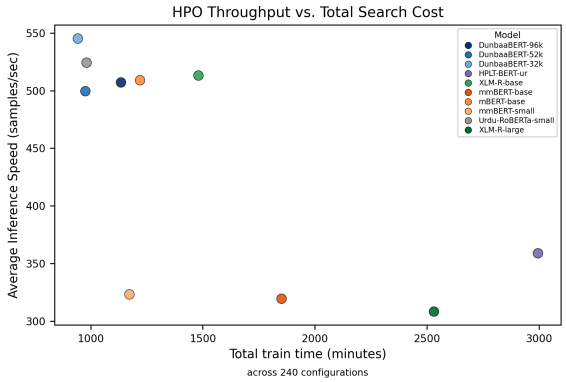
Table 7: For each model and benchmark, results are summarized over all 60 unique configurations, corresponding to 5 learning rates $\times$ 4 batch sizes $\times$ 3 random seeds, with early stopping using patience 3. Test Macro-F1, inference throughput in samples per second (SPS), and downstream fine-tuning time are reported as mean$_{\pm}$standard deviation. Norm. Eff. denotes normalized efficiency, computed within each benchmark from the corresponding mean Macro-F1 and mean SPS as $(\text{Macro-F1}/100) \times (\text{SPS}/\max(\text{SPS}))$. Higher values indicate a stronger performance–efficiency trade-off. Best results are shown in bold and second-best results are underlined.



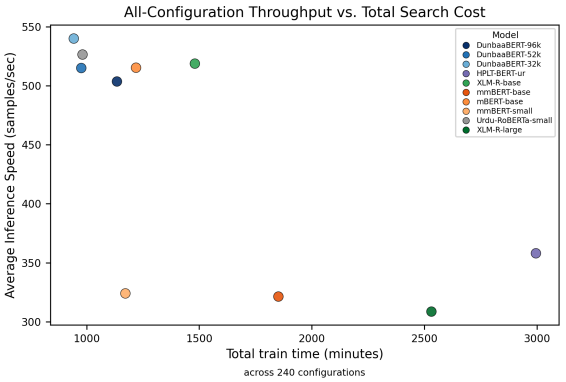
(a) HPO-selected Macro-F1 versus total search cost.



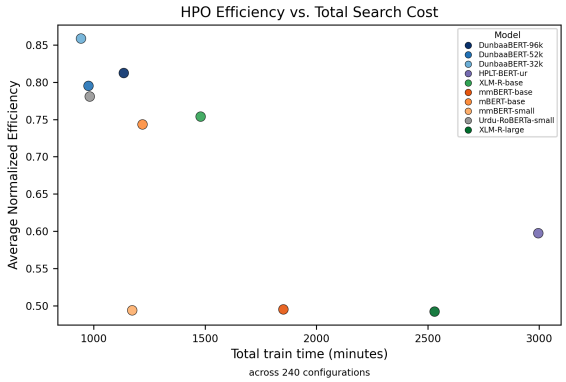
(b) All-configuration Macro-F1 versus total search cost.



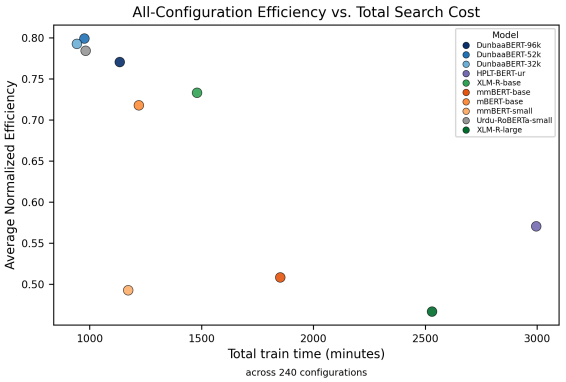
(c) HPO-selected inference throughput versus total search cost.



(d) All-configuration inference throughput versus total search cost.



(e) HPO-selected normalized efficiency versus total search cost.



(f) All-configuration normalized efficiency versus total search cost.

Figure 4: Aggregate relationship between total hyperparameter-search training cost and model performance–efficiency behavior across four downstream Urdu benchmarks (COUNT19, USADC, PSL–Kabbadi, IMDB-Urdu). Each point represents one model, and the x-axis reports total training time over the full unique search grid of 5 learning rates $\times$ 4 batch sizes $\times$ 3 seeds across 4 benchmarks, i.e., 240 configurations per model. The left column reports results after selecting the best HPO setting, while the right column summarizes behavior over all configurations. Panels (a,b) compare predictive performance using test Macro-F1, panels (c,d) compare inference throughput in samples per second, and panels (e,f) compare normalized efficiency (Norm. Eff.) **??**. Points closer to the upper-left region indicate a stronger trade-off between effectiveness and search cost.