MaiBERT: A Pre-training Corpus and Language Model for Low-Resourced Maithili Language

Sumit Yadav[1], Raju Kumar Yadav[1], Utsav Maskey[2], Gautam Siddharth Kashyap[2],

Ganesh Gautam[1], Usman Naseem[2]

[1]IOE, Pulchowk Campus, Lalitpur, Nepal

{076bct088.sumit, 076bct100.raju, ganesh}@pcampus.edu.np

[2]Macquarie University, Sydney, Australia

{utsav.maskey,usman.naseem}@mq.edu.au

gautam.kashyap@hdr.mq.edu.au

Abstract

Natural Language Understanding (NLU) for low-resource languages remains a major challenge in NLP due to the scarcity of high-quality data and language-specific models. Maithili, despite being spoken by millions, lacks adequate computational resources, limiting its inclusion in digital and AI-driven applications. To address this gap, we introduce **maiBERT**, a BERT-based language model pre-trained specifically for Maithili using the Masked Language Modeling (MLM) technique. Our model is trained on a newly constructed Maithili corpus and evaluated through a news classification task. In our experiments, **maiBERT** achieved an accuracy of 87.02%, outperforming existing regional models like NepBERTa and HindiBERT, with a 0.13% overall accuracy gain and 5–7% improvement across various classes. We have open-sourced **maiBERT** on Hugging Face[1], enabling further fine-tuning for downstream tasks such as sentiment analysis and Named Entity Recognition (NER).

1 Introduction

Natural Language Processing (NLP) has seen rapid advancements in recent years, with the development of sophisticated models like BERT (Devlin et al., 2019), GPT (Radford et al., 2018), and T5 (Raffel et al., 2020) that enable machines to understand, interpret, and generate human language. However, most of this progress has primarily focused on high-resource languages such as English, Mandarin, and Hindi. Low-resource languages—languages that lack sufficient digital content, annotated corpora, and computational resources—have remained largely underserved. One such language is Maithili[2] as shown in Fig. 1, an

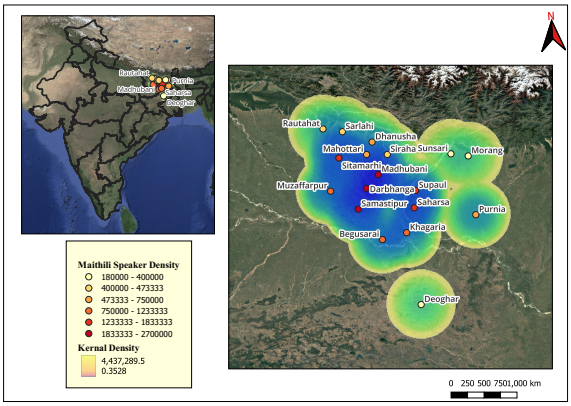

Figure 1: Density-Based Map of Maithili Language.

Indo-Aryan language spoken by millions of people in Nepal and India, particularly in the Terai region of Nepal (mainly Madhesh Province and parts of Koshi Province) and neighboring districts in the Indian state of Bihar.

Maithili is constitutionally recognized in Nepal and India and is widely spoken in both formal and informal contexts. Despite its rich linguistic and cultural heritage, Maithili has received minimal attention in the computational linguistics community (Mundotiya et al., 2021). Unlike related languages such as Hindi and Nepali, which have been the subject of substantial research (Joshi, 2022; Pudasaini et al., 2023; Maskey et al., 2022) and have several publicly available resources (Bal et al., 2024; Bafna and Žabokrtskỳ, 2022), Maithili lacks a comprehensive digital footprint. There are no large-scale pre-trained language models, standardized datasets, or open-source tools dedicated to this language, which significantly impedes the development of NLP applications for Maithili speakers. While existing models developed for related languages such as (Hartmann et al., 2017; Alian and Awajan, 2020; Pandit et al., 2019; Patil and Kolhe, 2022; Nie et al.,

[1]https://huggingface.co/rockerritesh/maiBERT_TF

[2]In terms of linguistic structure, Maithili follows a syntax and morphology that are comparable to other Devanagari-script languages such as Hindi and Nepali. It comprises 47 segmental phonemes: 8 vowel phonemes characterized by duration and quality, 39 consonant phonemes, and 10 numerals represented as letters.

2022; Ramchandra et al., 2019; Lalrempuii et al., 2021; Priyadarshi and Saha, 2020; Mundotiya et al., 2022) have shown some promise, they fall short when applied directly to Maithili due to dialectal variations, unique lexical items, and morphological structures that distinguish it from other Indo-Aryan languages.

To bridge this critical gap, we propose a Maithili-specific language model using the BERT architecture. Our work begins with the construction of a high-quality Maithili corpus collected from diverse sources, including online articles, books, and conversational data. This raw corpus forms the basis for pre-training a Masked Language Model (MLM) tailored specifically for Maithili. Following pre-training, we fine-tune the model for a downstream NLP task—text classification. Our model, referred to as **maiBERT**, is then benchmarked against State-of-the-Art (SOTA) to evaluate performance.

2 Related Works

NLP has shown remarkable advancements in foundational tasks such as semantic similarity and text categorization, significantly benefiting high-resource languages but posing persistent challenges for low-resource languages like Maithili (Chandrasekaran and Mago, 2021; Zhang et al., 2015). Therefore, to address the data scarcity in low-resource languages, transfer learning has emerged as a widely adopted strategy, where pretrained models from high-resource languages are fine-tuned on target languages (Ali et al., 2022). Patil and Kolhe (2022) demonstrated this by adapting an English sentiment model for Marathi through parameter optimization, although their study lacks scalability for languages with significantly different syntax and morphology. Similarly, Nie et al. (2022) proposed a novel cross-lingual emotion categorization model for Swahili and Urdu, but the absence of domain-specific datasets restricts real-world applicability. Ramchandra et al. (2019) explored semantic similarity in Hindi using multilingual deep learning models such as BERT and LASER, highlighting their comparative effectiveness; however, their results do not consider dialectal or regional variations commonly found in Hindi's linguistic relatives. Lalrempuii et al. (2021) developed a NER framework for Assamese using manually annotated data and neural networks, achieving reasonable accuracy; however, the manual annotation process is labor-intensive and limits its scalability. Priyadarshi and Saha (2020) created a Maithili POS tagger using CRF and neural word embeddings, reaching 85.88% accuracy with a 52,190-word corpus, but the system's performance may degrade for out-of-vocabulary or code-switched text. Mundotiya et al. (2022) developed annotated corpora and POS/chunking tools for Bhojpuri, Maithili, and Magahi, showing high accuracy for SAHBiLC and FineSAHBiLC models (e.g., 0.86%–0.95% for POS and chunking); still, their approach is limited by the static nature of the corpus and its lack of domain diversity. Word vectorization techniques, in general, have proven to be transformational for low-resource languages, yet the challenge remains in crafting embeddings that reflect the linguistic and morphological intricacies of underrepresented languages without relying excessively on sister-language transfer models. While transfer learning and distributional models enable rapid development, they often inherit biases and linguistic mismatches from their source languages.

3 Dataset

3.1 Pre-training Corpus:

With the objective of training language models for Maithili, we gathered approximately 18 million words by combining and de-duplicating data from three sources: (1) Maithili Wikipedia articles[3] ~880K words), (2) Digitized books[4] obtained through OCR processing of 418 scanned documents sourced from arXiv and other archives (~10.4 million words), and (3) raw newspaper articles, such as Esamaad[5], Maithilijindabad[6], and Maithili Purnajagran Prakash[7] (~6.6 million words). The OCR extraction was performed using Tesseract (Smith, 2007) on Maithili literary works and historical texts. After preprocessing and merging, the final corpus comprises over 450MB of cleaned Maithili text, making it one of the largest publicly compiled datasets for this language.

The combined raw corpus[8] for MLM pre-training consists of 1,028,017 sentences, with an average sentence length of 16.88 words and an average word length of 4.44 characters. Figure 2 provides an overview of the Maithili text corpus, il-

[3] https://mai.wikipedia.org
[4] https://archive.org/details/maithili-books/
[5] https://esamaad.com/
[6] https://maithilijindabaad.com/
[7] https://mppdainik.com/
[8] https://huggingface.co/datasets/rockerritesh/maithiliNewsData

lustrating how characters, words, and key linguistic features are distributed across the dataset.

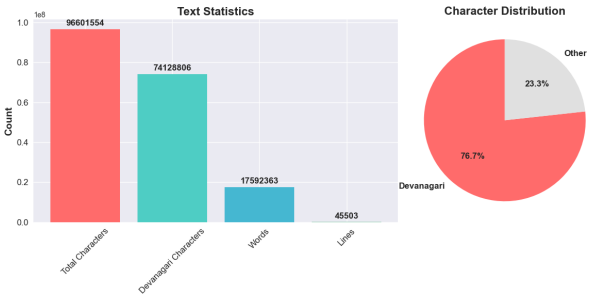

Figure 2: Comprehensive text statistics of the Maithili corpus showing character distribution, word counts, and linguistic features.

The corpus has a rich vocabulary of 1,357,081 unique tokens, highlighting its lexical diversity. A linguistic analysis revealed frequent occurrences of the Maithili-specific punctuation marker, with a total of 1,044,037 appearances across all text files—providing insights into syntactic structure and sentence boundaries.

Figure 3 displays the overall word distribution patterns in the Maithili text corpus.

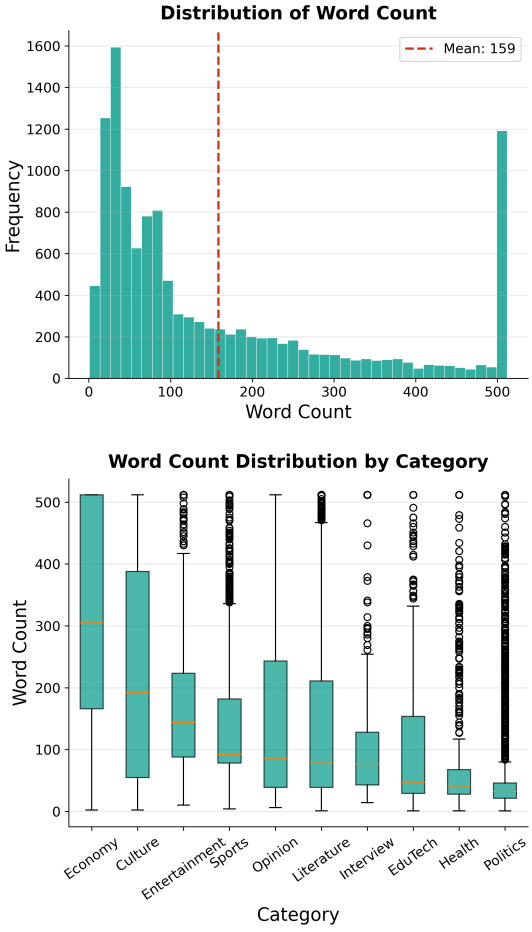

Figure 3: Word distribution patterns in Maithili corpus.

For enhanced language modeling, the raw data underwent extensive preprocessing, including normalization, lemmatization, and tokenization using a customized BertTokenizer[9], which preserved linguistic features specific to Devanagari script while filtering out non-Devanagari characters.

Classification Dataset: For downstream evaluation, we constructed a separate labeled test set from the Maithili news sources, as mentioned in the pre-training corpus.

The annotation process involved 3 native speakers and 2 linguistic experts who labeled the corpus into ten predefined categories, totaling 11,959 labeled instances, as shown in Figure 4. This distribution reflects the natural occurrence of topics in Maithili news media, with cultural and literary content being predominant.

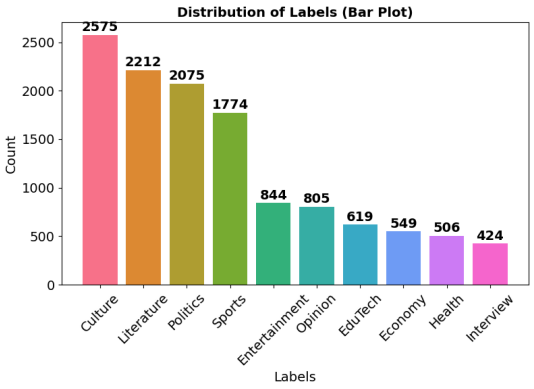

Figure 4: Distribution of news categories in the Maithili classification dataset.

To illustrate the varying levels of content complexity across categories, Figure 5 presents the distribution of word counts for different news sections.

Figure 6 shows the distribution of text lengths across the classification dataset, illustrating the variety in article lengths.

To validate the quality and consistency of annotations, inter-annotator agreement was computed using Cohen's Kappa coefficient, yielding a strong agreement score of 0.82, indicating reliable annotation practices. The labeled dataset was then split into training (70%), validation (15%), and test (15%) subsets to ensure robust performance evaluation. Table 1 provides a comprehensive overview of the classification dataset characteristics across different text complexity levels.

[9] https://www.tensorflow.org/text/api_docs/python/text/BertTokenizer

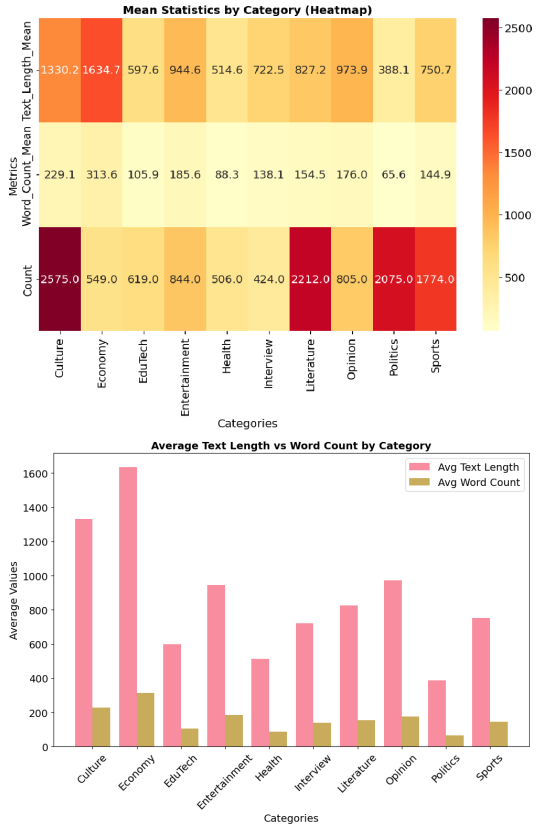

Figure 5: Word count distribution across different news categories in the classification dataset.

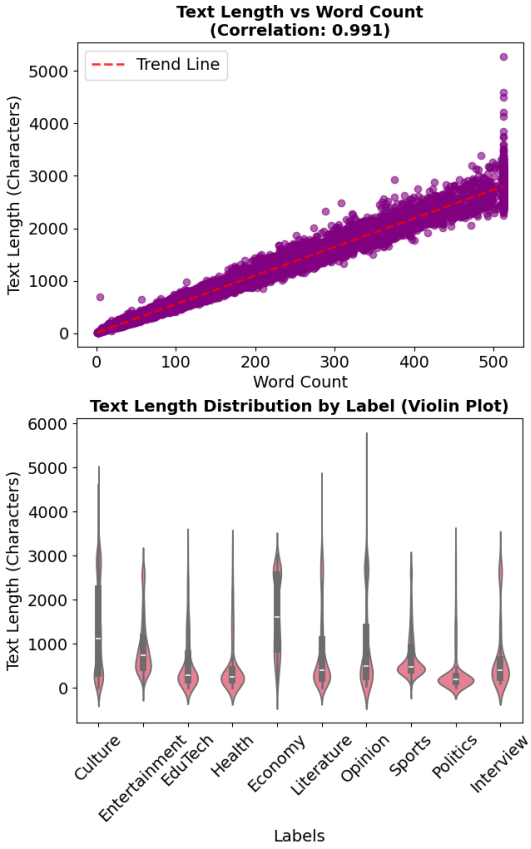

Figure 6: Distribution of text lengths in the Maithili classification dataset.

4 Methodology

In the development of **maiBERT**, a dedicated Transformer-based language model for the Mithila language, we adopt the TensorFlow implementation of the BERT[10] architecture, pretrained using the MLM objective. Given an input sequence $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, the model first maps each token x_i into a corresponding embedding $e_i \in R^d$, where d denotes the embedding dimension, forming an input matrix $E \in R^{n \times d}$. This matrix is combined with positional encodings $P \in R^{n \times d}$, ensuring sequential order preservation: $Z^{(0)} = E + P$. Each Transformer layer applies a multi-head self-attention mechanism, wherein attention heads h compute the scaled dot-product attention function as shown in Equation (1).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \quad (1)$$

where query, key, and value matrices $Q, K, V \in R^{n \times d_k}$ are derived via learned linear projections. Multiple heads capture diverse syntactic and semantic patterns, concatenated and projected to form the output $Z^{(l+1)}$. The stacked encoder layers include residual connections and layer normalization, enabling gradient flow across layers and mitigating vanishing gradients. The MLM training objective stochastically masks a subset of tokens x_i in the input sequence and optimizes the Negative Log-Likelihood (NLL) of correctly predicting the masked tokens as shown in Equation (2).

$$\mathcal{L}_{\text{MLM}} = -\sum_{i \in \mathcal{M}} \log P(x_i \mid x_{\mathcal{M}}) \quad (2)$$

where $\mathcal{M} \subseteq \{1, \ldots, n\}$ denotes the masked positions and $\mathbf{x}_{\mathcal{M}}$ represents the input sequence with masked tokens. We train a Mithila language Language model with text corpus sourced from `MithilaTextCorpus-v1` (refer to Section 3.1), ensuring a representative distribution of regional grammar, morphology, and vocabulary richness. The corpus was tokenized using `SentencePiece-BPE`, a subword-level tokenizer based on Byte-Pair Encoding (BPE), yielding a vocabulary $\mathcal{V}$ of size `30,000`, ensuring subword-level coverage of rare and compound words. Each training sample was tokenized into sequences of

[10] https://huggingface.co/docs/transformers/en/model_doc/bert

Category	Count	Avg Words	Avg Sent
Culture	2,575	147.4	8.5
Economy	549	176.8	8.5
EduTech	619	57.0	6.7
Entertainment	844	156.1	8.1
Health	506	58.5	4.3
Interview	424	95.4	6.8
Literature	2,212	155.0	10.3
Opinion	805	123.1	9.9
Politics	2,075	52.1	4.8
Sports	1,774	135.7	8.1
Total	**12,383**	**130.7**	**7.6**

Table 1: Dataset overview: Distribution and statistics across text categories for Count, Average number of words and Sentences.

length $n \leq 512$, padded or truncated appropriately. The final vocabulary embedding matrix $W_e \in R^{|\mathcal{V}| \times d} = R^{30000 \times 768}$ was learned jointly with model parameters. Training was conducted over `1,000,000` steps with a batch size of `64` and a learning rate $\eta = 5 \times 10^{-5}$, using the Adam optimizer with bias correction and weight decay regularization. Gradient updates followed the update rule according to Equation (3):

$$\theta_{t+1} \leftarrow \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{3}$$

where $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected estimates of the first and second moments of gradients. The model was implemented using TensorFlow's[11] API, allowing compatibility with Hugging Face's pretrained model interface. During inference or downstream fine-tuning, the model can be easily reloaded using TensorFlow's transformer ecosystem. For text generation or completion, masked token positions are inferred using Maximum a Posteriori (MAP) estimates. For classification tasks, the output hidden state $H \in R^{n \times d}$ from the final encoder layer is pooled using the CLS token representation $h_{[\text{CLS}]}$, then passed through a task-specific classification head $y = \text{softmax}(W_c h + b_c)$ with cross-entropy loss as shown in Equation (4).

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{4}$$

where C is the number of classes. Fine-tuning for sequence classification tasks employs the `TFAutoModelForSequenceClassification` interface, supporting task adaptation on custom Mithila language datasets. The model was trained on NVIDIA RTX 4090 GPUs with 24 GB VRAM using mixed-precision training (FP16) to reduce memory usage and accelerate matrix operations. Training leveraged TensorFlow's XLA compiler and `tf.keras.mixed_precision` policy for automatic loss scaling and efficient GPU utilization, resulting in improved throughput and reduced training time without compromising model accuracy. To mitigate overfitting, dropout regularization $\mathcal{D}(p = 0.1)$, early stopping with a patience of 3 epochs based on validation loss, and weight decay penalties $\lambda\|\theta\|^2$ with $\lambda = 0.01$ were applied. Token masking followed the BERT convention, with 15% of tokens masked: 80% replaced with `[MASK]`, 10% replaced with a random token, and 10% left unchanged. **maiBERT** embedding space offers potential for visualization and semantic similarity tasks, whereby cosine similarity $\cos(\theta) = \frac{a \cdot b}{\|a\|\|b\|}$ is used to quantify proximity in the learned semantic space. Algorithm 1 represents the entire flow of **maiBERT**.

5 Experimental Setup

5.1 Evaluation Metrics

To rigorously assess the performance of **maiBERT**, we employ a comprehensive suite of evaluation metrics suited for multi-class and imbalanced classification tasks. These include accuracy (%), macro-precision (%), macro-recall (%), and macro-F1-score (%).

5.2 Baselines

To assess the effectiveness of **maiBERT**, we compared its performance against a set of strong baseline models specifically designed for multilingual and low-resource Indic languages. These include NepBERTa (Timilsina et al., 2022), mBERT (Gonen et al., 2020), MuRIL (Khanuja et al., 2021), HindiBERT (Joshi, 2022), and NepaliBERT (Pudasaini et al., 2023). NepBERTa (Timilsina et al., 2022) is a RoBERTa-based model pre-trained on approximately 0.8 billion words of Nepali text, tailored to capture the linguistic nuances of the Nepali language. mBERT (Gonen et al., 2020), trained on Wikipedia data from 104 languages, covers 3.3 billion tokens and serves as a widely used multilingual benchmark. MuRIL (Khanuja et al., 2021) was pre-trained on a diverse mix of resources, including Wikipedia, translated corpora, OSCAR, and other Indian language datasets, comprising over 16

[11] `TFAutoModelForMaskedLM`

Algorithm 1 : **maiBERT**: Pretraining and Fine-tuning Procedure

Require: Corpus $\mathcal{C}$, vocabulary size $|\mathcal{V}| = 30000$, sequence length $n \leq 512$, embedding dimension $d = 768$, learning rate $\eta = 5 \times 10^{-5}$, batch size = 64, total steps $T = 1{,}000{,}000$

Ensure: Pretrained Transformer-based masked language model for Mithila language

Tokenization:
  Use `SentencePiece-BPE` to tokenize corpus $\mathcal{C}$ into subword units
  Build vocabulary $\mathcal{V}$ of size 30,000
  Convert sentences to token sequences of max length n, pad/truncate as necessary

Embedding Initialization:
  Initialize token embeddings $\mathbf{E} \in R^{n \times d}$ and positional encodings $\mathbf{P} \in R^{n \times d}$
  Form input to model: $\mathbf{Z}^{(0)} = \mathbf{E} + \mathbf{P}$

Training Loop:

for $t = 1$ to T **do**
  Sample batch of tokenized sequences $\mathbf{x} = [x_1, x_2, \ldots, x_n]$
  Randomly mask 15% of tokens to form $\mathbf{x}_{\mathcal{M}}$:
  80% replaced with `[MASK]`, 10% with random token, 10% unchanged
  Compute model output $\hat{x}_i$ for masked positions using MLM
  Compute MLM loss:

$$\mathcal{L}_{\text{MLM}} = -\sum_{i \in \mathcal{M}} \log P(x_i \mid \mathbf{x}_{\mathcal{M}})$$

  Compute gradients and update parameters using Adam optimizer:

$$\theta_{t+1} \leftarrow \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

  Apply regularization: weight decay $\lambda \|\theta\|^2$, dropout $\mathcal{D}(p)$

end for

Fine-Tuning:

if Downstream task is classification **then**
  Use output hidden states $\mathbf{H} \in R^{n \times d}$
  Extract `[CLS]` representation $h_{\text{[CLS]}}$
  Apply classification head:

$$\hat{y} = \text{softmax}(\mathbf{W}_c h + \mathbf{b}_c)$$

  Compute cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

  Train using `TFAutoModelForSequenceClassification`

end if

Inference:

Use MAP estimation for masked token prediction

Compute cosine similarity between embeddings for semantic tasks:

$$\cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$$

billion unique tokens. HindiBERT (Joshi, 2022) focuses on Hindi and Marathi, leveraging around 2.5 billion tokens from monolingual sources. NepaliBERT (Pudasaini et al., 2023) was trained on curated Nepali datasets, including the NepaliBERT (Pudasaini et al., 2023) corpus and additional Nepali-specific resources, encompassing approximately 0.8 billion tokens.

6 Results and Discussion

6.1 LLM Pre-training

We trained MaiBERT in the masked language modeling (MLM) fashion on the Mithila corpus (Section 3.1). Text was normalized (Devanagari-only, NFC), sentence-segmented, and tokenized with SentencePiece-BPE (|V| = 30,000). Inputs were truncated/padded to 512 tokens and masked following BERT's convention (15% masked: 80% [MASK], 10% random, 10% unchanged). We note down Training Loss and Perplexity in Table 2.

Epoch	Training Loss	Perplexity
1	0.45	1.23
3	0.37	1.01
6	0.21	1.00
10	0.012	1.00

Table 2: MLM pre-training progress showing convergence of loss and perplexity metrics across epochs.

Model	Maithali Classification Task			
Metric	Acc	Macro-Prec	Macro-Rec	Macro-F1
NepBERTa (Timilsina et al., 2022)	86.4	85.7	85.9	86.0
mBERT (Gonen et al., 2020)	55.2	54.5	53.8	54.1
MuRIL (Khanuja et al., 2021)	86.3	85.9	86.1	86.0
HindiBERT (Joshi, 2022)	**86.89**	**86.2**	**86.4**	**86.3**
NepaliBERT (Pudasaini et al., 2023)	82.0	81.5	81.2	81.3
maiBERT (Ours)	**87.02**	**86.8**	**87.0**	**86.9**

Table 3: Performance Comparison of Baseline Models on Text Classification Task

6.2 Downstream Classification and Comparison with SOTA

Table 3 demonstrates the performance of **maiBERT**. It handles Maithili news classification much better compared to existing state-of-the-art multilingual and Indic-specific baseline models. This consistent outperformance across all metrics can be attributed to several architectural and corpus-specific factors unique to **maiBERT**. First, unlike generic multilingual models such as mBERT (Gonen et al., 2020) and MuRIL (Khanuja et al., 2021), which are trained across a wide range of languages with varying resource availability, **maiBERT** is explicitly fine-tuned on a richly preprocessed and linguistically diverse Maithili corpus. This specialization enables it to better capture syntactic, semantic, and morphological nuances specific to the Maithili language, such as complex verb conjugations, dialectal variations, and usage of regional idioms, which are often underrepresented in large-scale multilingual pretraining. Moreover, the customized tokenization strategy leveraging a modified BERT tokenizer that preserves Devanagari-specific features plays a crucial role in improving embedding granularity, thereby enhancing context representation and classification robustness. The strong annotation quality and class-balanced labeled dataset further reinforce the model's learning capabilities, where high inter-annotator agreement (Cohen's Kappa = 0.82) ensures low label noise and improved generalization.

When compared to HindiBERT (Joshi, 2022) and NepBERTa (Timilsina et al., 2022)—both of which are tailored for Indic languages—**maiBERT** maintains a competitive edge due to its exclusive focus on Maithili language resources and domain-specific data collected from authentic news platforms. HindiBERT performs close to maiBERT (86.89% vs. 87.02%) on Maithili text, largely due to their shared Devanagari script, 60–70% lexical similarity, and standardized vocabulary in news domains. However, maiBERT outperforms HindiBERT by 5–7% in texts where Maithili-specific vocabulary and grammatical structures are more significant.

MaiBERT exhibits strong transfer capabilities and inherent language adaptability without needing extensive task-specific tuning. In contrast, mBERT (Gonen et al., 2020) and MuRIL (Khanuja et al., 2021) perform considerably lower, highlighting the limitations of overly generalized multilingual representations when applied to underrepresented and morphologically rich languages such as Maithili. Additionally, the inclusion of localized syntactic cues and lexical patterns during pretraining equips **maiBERT** with superior contextual disambiguation abilities, which proves critical in multi-class classification tasks where inter-class boundaries may be subtle.

6.3 Computational Experiments

The computational performance of **maiBERT** is evaluated using four key metrics: number of model parameters (Params, in billions), memory consumption during inference/training (Mem, in gigabytes), computational time (Time, in hours), and efficiency ratio (Ratio, in percentage). These metrics provide a holistic view of the model's resource utilization and optimization potential. Lower values of Params, Mem, and Time signify superior computational efficiency, while a higher efficiency Ratio reflects more optimal trade-offs between resource consumption and performance. As shown in Table 4, **maiBERT** maintains a compact architecture with just 0.11 billion parameters, resulting in lower memory usage of 4.6 GB. The computational time remains impressively low at 0.85 hours. Notably, the efficiency ratio is 172.11%, indicating that **maiBERT** delivers high predictive perfor-

mance relative to its computational cost. These results demonstrate the architectural and algorithmic efficiency of **maiBERT**, especially for resource-constrained environments typical of low-resource language modeling. The high efficiency ratio suggests that **maiBERT** leverages its compact parameter space and optimized training routines to maximize performance without imposing heavy memory or time burdens.

Model	Params (B)	Mem (GB)	Time (hrs)	Ratio (%)
maiBERT	0.11	4.6	0.85	172.11

Table 4: Computational Requirements of **maiBERT**. Ratio refers to Efficiency Ratio.

6.4 Traditional Machine Learning Baselines

To provide a comprehensive evaluation framework, we also compared **maiBERT** against traditional machine learning approaches on the same Maithili news classification task. Table 5 presents the performance of various classical ML algorithms including Support Vector Machines (SVM), Logistic Regression, Random Forest, Naive Bayes, and K-Nearest Neighbors, along with a Neural Network baseline. Among the traditional approaches, SVM with linear kernel achieved the highest accuracy of 76.70% and cross-validation accuracy of 75.63%, followed by SVM with RBF kernel (75.73% accuracy). The Neural Network baseline, comprising multiple fully-connected layers with batch normalization and dropout regularization, achieved 72.62% test accuracy. These results demonstrate that while traditional ML methods can achieve reasonable performance on Maithili text classification, they fall significantly short of the 87.02% accuracy achieved by **maiBERT**, highlighting the effectiveness of transformer-based architectures and language-specific pretraining for low-resource language understanding tasks.

Model	Accuracy	CV Accuracy
SVM (Linear)	76.70	75.63
SVM (RBF)	75.73	74.65
Logistic Regression	74.92	73.71
Random Forest	65.63	64.83
Naive Bayes	64.58	63.34
K-Nearest Neighbors	21.12	19.15
Neural Network	72.62	–

Table 5: Performance comparison of traditional ML approaches on Maithili news classification task and Cross-validation accuracy.

7 Conclusion and Future Works

In this work, we introduced **maiBERT**, a transformer-based language model tailored for the Maithili language, establishing a strong foundation for NLU tasks. Our model demonstrated competitive performance across baselines. By leveraging a carefully curated Maithili corpus, we addressed critical gaps in low-resource language processing and highlighted the importance of linguistic inclusivity in NLP research. Despite its effectiveness in understanding tasks, limitations remain in generative capabilities and domain adaptability, particularly in creative or poetic contexts. Future work will focus on expanding the diversity of the training corpus to include more conversational, literary, and informal texts. Moreover, we aim to enhance the model's generative capacity through causal language modeling and integrate instruction tuning for downstream applications.

References

Raza Ali, Umar Farooq, Umair Arshad, Waseem Shahzad, and Mirza Omer Beg. 2022. Hate speech detection on twitter using transfer learning. *Computer Speech & Language*, 74:101365.

Marwah Alian and Arafat Awajan. 2020. Semantic similarity for english and arabic texts: a review. *Journal of Information & Knowledge Management*, 19(04):2050033.

Niyati Bafna and Zdeněk Žabokrtskỳ. 2022. Subword-based cross-lingual transfer of embeddings from hindi to marathi and nepali. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 61–71.

Bal Krishna Bal, Balaram Prasain, Rupak Raj Ghimire, and Praveen Acharya. 2024. Strategies for corpus development for low-resource languages: Insights from nepal. *Automatic Speech Recognition and Translation for Low Resource Languages*, pages 297–330.

Dhivya Chandrasekaran and Vijay Mago. 2021. Evolution of semantic similarity—a survey. *Acm Computing Surveys (Csur)*, 54(2):1–37.

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.

Hila Gonen, Shauli Ravfogel, Yanai Elazar, and Yoav Goldberg. 2020. It's not greek to mbert: inducing

word-level translations from multilingual bert. *arXiv preprint arXiv:2010.08275*.

Nathan Hartmann, Erick Fonseca, Christopher Shulby, Marcos Treviso, Jessica Rodrigues, and Sandra Aluisio. 2017. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *arXiv preprint arXiv:1708.06025*.

Raviraj Joshi. 2022. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, and 1 others. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Candy Lalrempuii, Badal Soni, and Partha Pakray. 2021. An improved english-to-mizo neural machine translation. *Transactions on Asian and Low-Resource Language Information Processing*, 20(4):1–21.

Utsav Maskey, Manish Bhatta, Shiva Bhatt, Sanket Dhungel, and Bal Krishna Bal. 2022. Nepali encoder transformers: An analysis of auto encoding transformer language models for Nepali text classification. In *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 106–111, Marseille, France. European Language Resources Association.

Rajesh Kumar Mundotiya, Swasti Mishra, and Anil Kumar Singh. 2022. Hierarchical self attention based sequential labelling model for bhojpuri, maithili and magahi languages. *Journal of King Saud University-Computer and Information Sciences*, 34(10):8739–8749.

Rajesh Kumar Mundotiya, Manish Kumar Singh, Rahul Kapur, Swasti Mishra, and Anil Kumar Singh. 2021. Linguistic resources for bhojpuri, magahi, and maithili: statistics about them, their similarity estimates, and baselines for three applications. *Transactions on Asian and Low-Resource Language Information Processing*, 20(6):1–37.

Ercong Nie, Sheng Liang, Helmut Schmid, and Hinrich Schütze. 2022. Cross-lingual retrieval augmented prompt for low-resource languages. *arXiv preprint arXiv:2212.09651*.

Rajat Pandit, Saptarshi Sengupta, Sudip Kumar Naskar, Niladri Sekhar Dash, and Mohini Mohan Sardar. 2019. Improving semantic similarity with cross-lingual resources: a study in bangla—a low resourced language. In *Informatics*, volume 6, page 19. MDPI.

Rupali S Patil and Satish R Kolhe. 2022. Supervised classifiers with tf-idf features for sentiment analysis of marathi tweets. *Social Network Analysis and Mining*, 12(1):51.

Ankur Priyadarshi and Sujan Kumar Saha. 2020. Towards the first maithili part of speech tagger: Resource creation and system development. *Computer Speech & Language*, 62:101054.

Shushanta Pudasaini, Subarna Shakya, Aakash Tamang, Sajjan Adhikari, Sunil Thapa, and Sagar Lamichhane. 2023. Nepalibert: Pre-training of masked language model in nepali corpus. In *2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pages 325–330. IEEE.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training.

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

Joshi Ramchandra, Purvi Goel, and Raviraj Joshi. 2019. Deep learning for hindi text classification: a comparison. In *International Conference on Intelligent Human Computer Interaction*. Springer.

R. Smith. 2007. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633.

Sulav Timilsina, Milan Gautam, and Binod Bhattarai. 2022. Nepberta: Nepali language model trained in a large corpus. In *Proceedings of the 2nd conference of the Asia-pacific chapter of the association for computational linguistics and the 12th international joint conference on natural language processing*. Association for Computational Linguistics (ACL).

Shuang Zhang, Xuefeng Zheng, and Changjun Hu. 2015. A survey of semantic similarity and its application to social network analysis. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2362–2367. IEEE.