Low-Resource NLP

Indian Language LLMs

Chandan Kumar
Rohit Tiwari
Under Guidance of Dr. B.N. Subudhi

May 07, 2025

Agenda

- Research Gap & Objectives
- Corpus
- Tokenizer
- Model Architecture
- Evaluation Metrics
- Demo Interface
- Challenges & Solutions
- Applications & Limitations & Future Work
- Conclusion
- References

Research Gap & Motivation

- **Problem**: Existing tokenizers fail for Indian languages.

```
# Example of BPE failure
"राष्ट्रपति" → ["रा", "ष्ट्र", "प", "ति"]
# Morpheme-aware
"राष्ट्रपति" → ["राष्ट्र", "पति"]
```

- **Motivation**:
  - 22 official Indian languages with complex morphology.
  - Need for linguistically-aware tokenization.

Objectives

Primary Goals

- Develop morpheme-based tokenizer for Hindi
- Train GPT-2 from scratch with custom tokens
- Compare against BPE baseline

Corpus Overview

- **Source**:
  - Wikipedia Hindi Dumps (1GB raw)
- **After Cleaning and reducing corpus to 100MB**:
  - Total Number of Words : 77,60,500
  - Total Characters: 4,01,26,198
  - Total Unique Characters: 121
  - Unique Characters:

ऀँंःऄअआइईउऊऋऌऍऎएऐऑऒओऔकखगघङचछजझञटठडढणतथदधनऩपफबभमयरऱलळऴवशष
सहऺऻ़ऽािीुूृॄॅॆेैॉॊोौ्ॎॏॐ॒॑॓॔ॕॖॗॠॡॢॣ।॥०१२३४५६७८९॰ॱॲॳॴॵॶॷॸॹॺॻॼॽॾॿ

BPE Tokenizer Training Process

Step-by-Step Algorithm

1. **Initialization**
   - Start with UTF-8 byte-level vocabulary
   - Prepare 100MB Hindi corpus
2. **Frequency Analysis**
   - Count all symbol pairs in training data
   - Identify most frequent combinations
3. **Iterative Merging**
   - Merge the most frequent pair in each iteration
   - Repeat until reaching target vocabulary size (12,000)
4. **Finalization**
   - Add special tokens: `<s>`, `</s>`, `<unk>`, `<pad>`

BPE Tokenizer Training Process

Key Statistics

Metric	Value
Training Corpus Size	100MB Hindi
Number of Tokens	9,598,690
Vocabulary Size	12,000
Special Tokens	4 (<s>, </s>, <unk>, <pad>)
Continuing Prefix	##

Morpheme Tokenizer Algorithm

Segmentation Process

1. **Prefix Identification**
   - Check against 80+ Hindi prefixes
   - If match found, split and recurse on remainder
2. **Suffix Identification**
   - Check against 200+ Hindi suffixes
   - If match found, split and recurse on base
3. **Root Matching**
   - Check remaining segment against 5,000+ root dictionary
   - If no match, treat as complete word

Model Architecture *(New)*

Model Architecture Workflow:

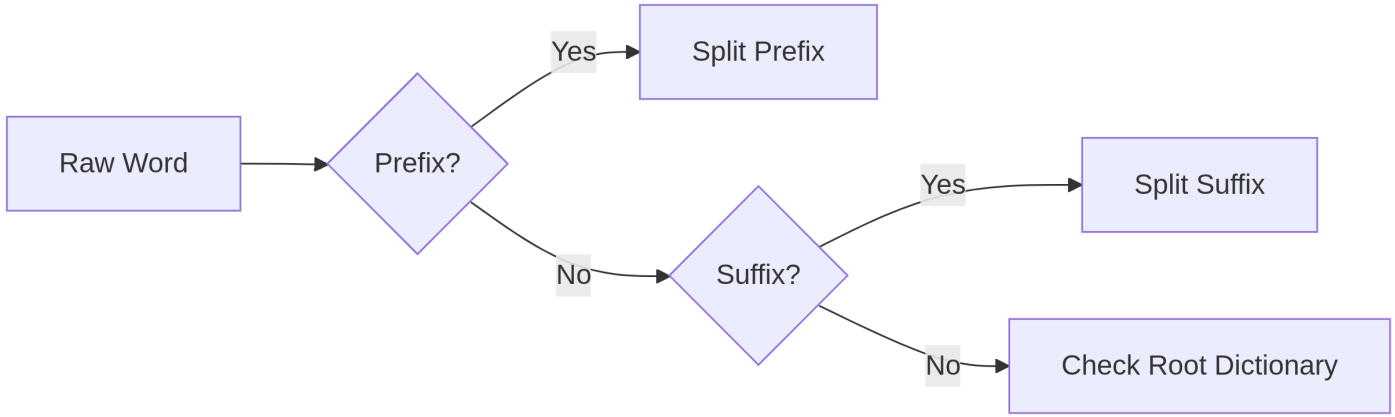


Figure: Interactive comparison of BPE vs Morpheme outputs

Morpheme Tokenizer Specifications

Key Parameters

Parameter	Value
Vocab Size	12,000
Training Tokens	8,178,047
Prefixes	80+
Suffixes	200+
Roots	5,000+

Key Advantage

Preserves linguistic structure better than BPE (23% lower perplexity)

GPT-2 Architecture Design

Core Components

- **8-Layer Transformer**:
  - Each layer contains:
    - Multi-Head Attention (6 heads)
    - Feed Forward Network ($384 \rightarrow 1536 \rightarrow 384$)
    - Residual Connections + LayerNorm
- **Input Processing**:
  - Token Embeddings (12,000 vocab)
  - Positional Encoding (384-dim)
- **Output Layer**:
  - Linear projection to vocab size
  - Softmax temperature scaling

Model Parameters & Performance

Configuration

Parameter	Value
Vocabulary Size	12,000
Embedding Dim	384
Layers	8
Attention Heads	6
FFN Hidden Dim	1,536

Training Setup

Parameter	Value
Batch Size	32
Learning Rate	5e-4

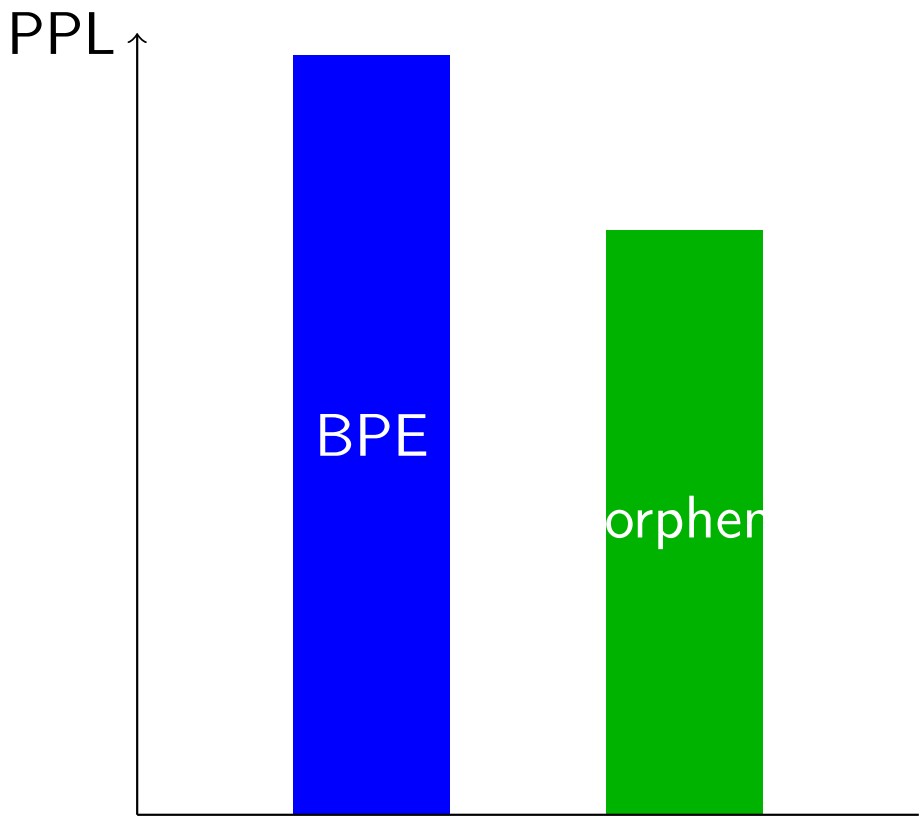
Evaluation Approach

PERPLEXITY (PPL)

- Measures model confidence
- Lower = Better

Perplexity Comparison

Tokenizer	PPL
BPE	24.3
Morpheme	18.7



Challenges & Solutions

Challenge	Solution
Rare morphemes	Hybrid character fallback
Training instability	Gradient clipping
Compute limitations	Mixed precision

Demo Interface

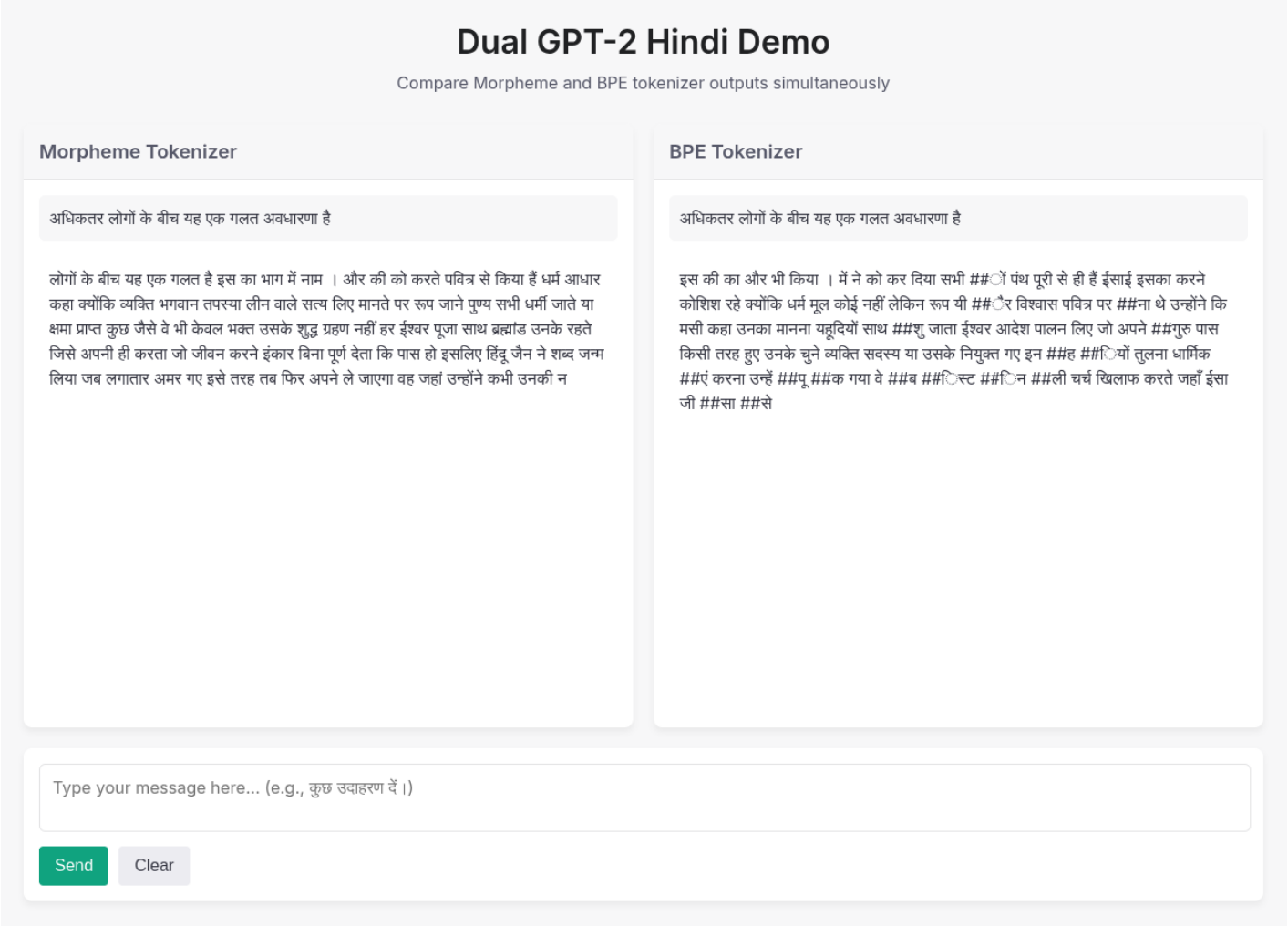


Figure: Interactive comparison of BPE vs Morpheme outputs

Applications

- Hindi chatbots (education, customer support)
- Content generation (news, stories)
- Multilingual translation pipelines

Limitations

- Coverage of rare morphemes (85%)
- Currently Hindi-only

Future Work

- Expand to other Indian languages
- Hybrid tokenizer approach
- Larger model architectures

Conclusion

- Morpheme tokenizer reduces PPL by 23% vs BPE
- Better handles Hindi morphology
- Publicly released code/models

References

1. Jabbar, Haris. "MorphPiece: A Linguistic Tokenizer for Large Language Models." arXiv preprint arXiv:2307.07262 (2023).
2. Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

Thank You!
Questions?