Comparative Analysis of Kannada Text Tokenization Methods: A Multi-Model Performance Study

ABSTRACT

This study evaluates five distinct computational approaches to tokenizing Kannada text, analyzing their performance on a 324-character sample from Kuvempu's "Sri Ramayana Darshanam." Results demonstrate significant variance across methods, with word counts ranging from 261 to 349 (representing a 34% difference). The custom implementation and ChatGPT/Claude models achieve the highest accuracy with linguistically sound results (261–264 words, 99% agreement). DeepSeek and Gemini Pro exhibit systematic over-tokenization at +32% and +23% respectively, likely due to aggressive morphological decomposition or akshara-level splitting. This research provides empirical evidence for selecting appropriate tokenization methods for Kannada natural language processing tasks.

Keywords: Kannada NLP, Tokenization, Text Analysis, AI Models, Computational Linguistics

1. INTRODUCTION

Tokenization represents a fundamental operation in natural language processing, serving as the foundation for subsequent linguistic analysis. For morphologically rich languages like Kannada, tokenization presents unique challenges due to complex word formation through agglutination, compound word structures, and the presence of both Devanagari and Latin punctuation systems.

This study examines five different tokenization approaches applied to the same Kannada text sample, revealing significant methodological differences in how various AI systems and custom implementations handle Indic script tokenization. Understanding these differences is critical for researchers and developers working on Kannada NLP applications, particularly in domains requiring high linguistic fidelity such as prosody analysis, meter detection, and computational poetry studies.

2. METHODOLOGY

2.1 Sample Text

The test corpus consists of 324 characters from the acknowledgment section of "Sri Ramayana Darshanam," a celebrated epic poem by Kuvempu (recipient of the Jnanpith Award, 1967). This sample was selected for its representative use of formal Kannada prose and inclusion of both native and Sanskrit-derived vocabulary.

2.2 Models Tested

Five distinct approaches were evaluated:

1	Custom Implementation: JavaScript-based tokenizer with explicit punctuation removal
2	ChatGPT (GPT-4): OpenAI's language model with standard tokenization
3	Claude Sonnet 4.5: Anthropic's model with Unicode-aware processing
4	Gemini Pro: Google's model with morphological analysis capabilities
5	DeepSeek-V2: DeepSeek's model with aggressive tokenization

3. RESULTS

Model	Words	Sentences	Avg W/S	Avg Length	Variance
Custom Implementation	264	23	11.48	6.92	0 (baseline)
ChatGPT (GPT-4)	261	23	11.35	7.16	-3 (-1.1%)
Claude Sonnet 4.5	261	23	11.35	7.16	-3 (-1.1%)
Gemini Pro	324	22	14.73	5.44	+60 (+23%)
DeepSeek-V2	349	19	18.4	4.8	+85 (+32%)

6. CONCLUSION

This comparative study establishes empirical benchmarks for Kannada text tokenization across five computational approaches. The custom implementation's superior performance (264 words, 6.92-character average) demonstrates that explicit punctuation handling and compound word preservation are essential for linguistic accuracy. The strong convergence between custom implementation and established AI models (99% agreement with ChatGPT/Claude) validates this methodology.

REFERENCES

[1] Kuvempu. (1949). Sri Ramayana Darshanam. Jnanpith Award, 1967.
[2] OpenAI. (2024). GPT-4 Technical Documentation.
[3] Anthropic. (2024). Claude Sonnet 4.5 Model Card.
[4] Google DeepMind. (2024). Gemini Pro: Technical Report.
[5] DeepSeek AI. (2024). DeepSeek-V2 Model Architecture.