ENPM 703 Final Project Milestone Report

Kirti Kishore, Tanmay Pancholi, Suraj Kalwaghe
University of Maryland
Master's of Engineering (Robotics)
{kiki1, tamy2909, suraj108}@umd.edu

Abstract

This project develops an AI model to interpret ancient Hindu Vedic scriptures and provides personalized life guidance based on Vedic wisdom. The goal is to bridge ancient knowledge with modern technology, enhancing accessibility and applicability to contemporary lifestyles.

1. Introduction

In this project, we aim to develop a Natural Language Processing (NLP) model that can derive insights from ancient Hindu Vedic scriptures, particularly focusing on Sanskrit texts. The goal is to offer personalized life guidance based on these texts, such as advice on diet, sleep, and auspicious activities. Given the complexity and richness of Sanskrit, this task presents significant challenges in terms of tokenization, understanding, and generating relevant advice. Our approach involves comparing various language models, including MuRIL, BERT, SanBERT, and ByT5, to evaluate their performance in tokenizing Sanskrit texts effectively. The primary metric for comparison is the fertility score, which measures the number of tokens generated per word. Lower fertility scores indicate better tokenization efficiency, which is crucial for handling morphologically rich languages like Sanskrit.

2. Problem Statement

The problem we are addressing is the development of an AI model capable of interpreting and deriving insights from ancient Hindu Vedic scriptures, particularly in Sanskrit, to offer personalized life guidance. This guidance could cover various aspects of life, such as diet, sleep, and auspicious activities, based on the wisdom contained in these texts. The challenge lies in translating ancient Vedic knowledge into practical advice for modern users using Natural Language Processing (NLP) techniques.

2.1. Dataset

The dataset comprises digital versions of the Vedas and other Hindu scriptures, specifically focusing on Sanskrit texts. These texts are sourced from verified online repositories to ensure authenticity and quality. The texts are provided in Sanskrit, a morphologically rich and complex language with unique linguistic structures. The dataset requires preprocessing steps such as tokenization, sandhi splitting (to handle compound words), and normalization to prepare it for NLP tasks.

2.2. Input/Output of the Model

For intermediate evaluation of our project, the input to the model will be raw Sanskrit text from the Vedas or related scriptures. The output will be tokenized text and ultimately towards the end oof our project we aim the output to be personalized life guidance generated from the Vedic teachings. This guidance will be provided in natural language based on user queries.

2.3. Evaluation Metric

For this milestone, the primary evaluation metric is the fertility score, which measures tokenization efficiency. Fertility is defined as the average number of tokens generated per word by a tokenizer. Lower fertility scores indicate more efficient tokenization, which is particularly important for morphologically rich languages like Sanskrit. In later stages, additional metrics such as accuracy, BLEU score (for text generation), and user feedback will be used to evaluate the model's performance in generating meaningful life guidance.

2.4. Expected Results

We expect models specifically trained on Indian languages (e.g., MuRIL) or fine-tuned for Sanskrit (SanBERT) to outperform general-purpose models like BERT in terms of tokenization efficiency. Byte-level models like ByT5 may struggle with efficiency due to their approach of breaking down text into individual bytes rather than words or subwords.

2.5. Difference from State-of-the-Art/Baseline Implementation

While many NLP models have been applied to modern languages or religious texts like the Bible or Quran, few have been tailored specifically for Sanskrit or Vedic texts. Our approach differs by focusing on models that are either trained specifically for Indian languages (MuRIL) or fine-tuned for Sanskrit (SanBERT). Additionally, we compare these models with byte-level models like ByT5 to understand how different tokenization strategies affect performance on Sanskrit texts.

3. Literature Review

3.1. Advancements in Multilingual Models for Sanskrit

Recent advancements in NLP, such as MuRIL and San-BERT, are specifically designed to address the complexities of Sanskrit and other Indian languages. MuRIL excels in handling the intricate morphologies and diverse scripts, enhancing performance in tasks like named entity recognition and question answering. San-BERT, developed using a tailored Sanskrit corpus, focuses on extractive summarization, significantly improving the accessibility and understanding of Sanskrit texts through advanced NLP techniques.

3.2. Thematic Consistency in Hindu Philosophy Texts

Using BERT-based models, Chandra and Ranjan conducted topic modelling on the Upanishads and Bhagavad Gita, demonstrating a high degree of thematic consistency. Their findings reveal a mean cosine similarity of 73 percent, underscoring the potential of AI to uncover and articulate the nuanced themes of these ancient scriptures effectively.

3.3. Neural Word Embeddings in Sanskrit Processing

Sandhan et al. explored the impact of both static and contextualized neural word embeddings on Sanskrit text processing. Their research highlighted that while static embeddings effectively capture syntactic nuances, contextualized embeddings like ELMo and ALBERT provide superior semantic understanding by adapting to the contextual variability inherent in the language.

3.4. Semantic and Sentiment Analysis Across Translations

Chandra and Kulkarni's work on semantic and sentiment analysis of the Bhagavad Gita translations using a BERT-based framework illustrates how deep learning models can maintain philosophical and ethical integrity across different translations. Their analysis confirms the consistency of underlying messages and sentiments across versions, showcasing the robustness of NLP tools in maintaining the essence of the text despite varied linguistic expressions.

4. Technical Approach

To solve the problem of efficiently tokenizing and processing Sanskrit texts for personalized life guidance, we compare four different language models: MuRIL, BERT, SanBERT, and ByT5. Each model has its own strengths and weaknesses in handling morphologically rich languages like Sanskrit.

4.1. Data Preprocessing

Sanskrit texts are tokenized using each model's tokenizer to evaluate how well each model handles the complex morphology and rich inflectional system of Sanskrit. Texts are cleaned and normalized by removing unnecessary characters or symbols that may interfere with tokenization. Sandhi (word junctions) in Sanskrit often results in compound words, which need to be split appropriately during preprocessing to ensure accurate tokenization.

4.2. Models Compared

MuRIL (Multilingual Representations for Indian Languages) is specifically trained on Indian languages, including Sanskrit. It uses a multilingual BERT architecture and is expected to perform well on Indian languages due to its pretraining on large corpora of Indian text. BERT(Base Multilingual Cased) is a general-purpose multilingual model trained on 104 languages but not specifically fine-tuned for Indian languages or Sanskrit. We expect it to perform less efficiently than models specifically trained on Indian languages. SanBERT is a BERT-based model fine-tuned specifically for Sanskrit. This model is expected to handle the complexities of Sanskrit better than general-purpose models due to its fine-tuning on Sanskrit texts. ByT5 is a byte-level model that tokenizes text at the byte level rather than at the word or subword level. It is expected to generate more tokens per word, leading to higher fertility scores, but it may struggle with tokenization efficiency for morphologically rich languages like Sanskrit.

4.3. Results

The fertility score is calculated as the average number of tokens generated per word by each tokenizer. For each model, we tokenize the input Sanskrit texts and calculate the fertility score. Lower fertility scores indicate better tokenization efficiency, which is crucial for handling languages like Sanskrit that have complex word structures. On implementing the fertility check with all the pre-trained models with our manual dataset we got the plotting for fertility

scores. The following bar chart shows the comparison of fertility scores across different models for tokenizing Sanskrit texts.

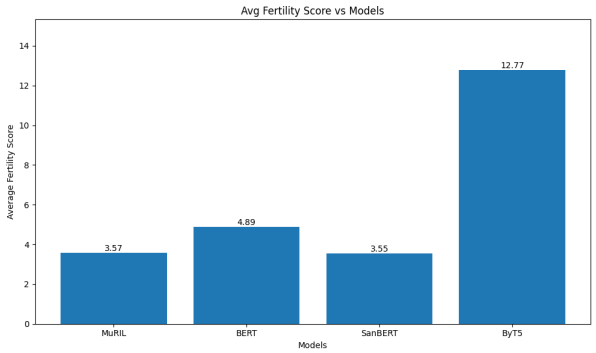

Figure 1. Average Fertility Score vs Model

4.4. Analysis of Results

San-BERT and MuRIL have the lowest fertility scores (3.55 and 3.57 respectively), indicating that they are more efficient at tokenizing Sanskrit texts compared to BERT and ByT5. This result is expected as both San-BERT and MuRIL are either fine-tuned or specifically trained on Indian languages, including Sanskrit. BERT, being a general-purpose multilingual model, has a higher fertility score (4.89), suggesting that it struggles more with handling the complexities of Sanskrit compared to San-BERT and MuRIL. ByT5, with its byte-level tokenization approach, generates significantly more tokens per word (fertility score of 12.77). While byte-level models can handle any language at a byte level, this approach results in much higher token counts for morphologically rich languages like Sanskrit. The lower fertility scores for MuRIL and San-BERT suggest that these models are better suited for tasks involving Sanskrit text processing compared to general-purpose models like BERT or byte-level models like ByT5. ByT5's high fertility score indicates that it may not be ideal for tasks requiring efficient tokenization of complex languages like Sanskrit, where morphological richness leads to an explosion in token count when using byte-level tokenization.

5. Conclusion

For downstream tasks such as question-answering or generating life guidance based on Vedic texts, models like San-BERT and MuRIL are more likely to perform better due to their ability to tokenize Sanskrit efficiently. Further evaluation will involve testing these models on specific tasks such as question-answering or text generation to see how their tokenization efficiency translates into task performance.

6. Next Steps

Fine-tune San-BERT and MuRIL further using specific Vedic texts from other scriptures such as the Rig Veda or Upanishads to improve their understanding of Vedic concepts. Implement downstream tasks such as question-answering or generating personalized life guidance based on user queries related to Vedic teachings. In addition to fertility scores, evaluate these models using task-specific metrics such as accuracy, BLEU score (for text generation), perplexity (for language modeling), and user feedback. Develop a user-friendly interface that allows users to input queries related to Vedic teachings and receive personalized life guidance based on the output from these NLP models.
[3] [6] [5] [1] [2] [4]

References

[1] K. Bhatnagar, S. Lonka, J. Kunal, and M. R. M. G. San-bert: Extractive summarization for sanskrit documents using bert and it's variants, 2023. 3

[2] R. Chandra and V. Kulkarni. Semantic and sentiment analysis of selected bhagavad gita translations using bert-based language framework. *CoRR*, abs/2201.03115, 2022. 3

[3] R. Chandra and M. Ranjan. Artificial intelligence for topic modelling in hindu philosophy: Mapping themes between the upanishads and the bhagavad gita. *PLOS ONE*, 17(9):1–34, 09 2022. 3

[4] B. Hutchinson. Modeling the sacred: Considerations when using religious texts in natural language processing, 2024. 3

[5] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, S. Gupta, S. C. B. Gali, V. Subramanian, and P. Talukdar. Muril: Multilingual representations for indian languages, 2021. 3

[6] J. Sandhan, O. Adideva, D. Komal, L. Behera, and P. Goyal. Evaluating neural word embeddings for sanskrit, 2021. 3