Bosubabu Sambana

Web-page: https://github.com/bosubabu/Bosubabu-Sambana#
https://www.bosubabu.com/papers/journals

Project Overview

Bosubabu Sambana is an initiative aimed at developing an English-to-Sanskrit sentence translation model. While translation technologies are widely used across industries—from business and media to government and social networks—there remains a significant gap when it comes to high-quality, automated Sanskrit translation tools. Existing resources largely focus on word-by-word dictionaries and do not account for sentence-level grammar or context.

According to the 2001 Indian census, around five million people speak Sanskrit as either a first, second, or third language. Despite this, there is a lack of robust tools for full-sentence translation to and from Sanskrit. Creating such a system would provide valuable utility across various sectors and for individual users alike.

Machine Translation (MT) is a well-established area within data science, with numerous models already applied successfully to various language pairs. Adapting these models for Sanskrit, especially given its unique grammatical structure and script, presents both a challenge and an opportunity. Fortunately, the availability of the International Alphabet of Sanskrit Transliteration (IAST), which uses Latin script, can help reduce some of the complexity involved in working with Devanagari, the standard script for Sanskrit.

Project Objectives

The primary aim is to build a machine learning model capable of translating between multiple languages, with a final focus on English-Sanskrit translation. The project will begin with more widely studied language pairs, such as English-German, to help the model grasp fundamental linguistic patterns. These initial steps will serve as a foundation before progressing to Sanskrit, a language with far less existing translation support.

By starting with common language pairs, we can validate our model against established benchmarks and ensure its performance is reliable. Throughout the process, we will split the dataset into training and test sets to systematically evaluate the model's effectiveness.

Modeling Approach

Our core model will be based on the **Encoder-Decoder architecture using Long Short-Term Memory (LSTM)** networks. LSTM is a type of recurrent neural network (RNN) that excels at learning from sequences where both short-term and long-term dependencies are important—making it ideal for language translation.

- **Encoder:** Processes and encodes the input sentence into a context vector that captures its semantic meaning.
- **Decoder:** Takes the encoded vector and generates the translated sentence, word by word or character by character.

This structure is particularly effective because it allows the model to retain contextual understanding throughout the translation process.

Another method under consideration is **Statistical Machine Translation (SMT)**, which builds translation models based on statistical relationships found in bilingual corpora. For Sanskrit, where morphology plays a crucial role, we plan to integrate morphological analysis tools to enhance translation accuracy. Tools like **Morphogen** can aid in handling the morphological richness of Sanskrit compared to less inflectional languages like English.

Technologies and Tools

We plan to use **TensorFlow**, a popular open-source library in Python, for building and training our models. TensorFlow offers a user-friendly interface, extensive documentation, and powerful features like GPU acceleration via CUDA, which will be useful given the computational demands of our models.

We may also explore other frameworks such as **MXNet**, which supports multi-GPU training and might provide performance advantages for large datasets and more complex models.

This project blends established translation techniques with the unique challenges posed by Sanskrit, aiming to fill a gap in current translation tools by building a comprehensive and context-aware translation system.

Data Set:

1. **INRIA Sanskrit Manual**
   Website: http://sanskrit.inria.fr/manual.html
   This platform provides tools and resources for Sanskrit morphological tagging. The morphological data available here is valuable for use in Statistical Machine Translation (SMT), where accurate word analysis is crucial.

2. **Sanskrit Bible (New Testament Translation)**
   Website: http://sanskritbible.in
   This resource suggestions aligned Sanskrit and English verses from the New Testament. The parallel text pairs can be extracted as JSON objects, which is particularly useful for training sequence-to-sequence models such as LSTM-based translators.

3. **OPUS Project**
   Website: http://opus.nlpl.eu
   A comprehensive repository of multilingual parallel corpora including English, German,

French, Spanish, and other languages. While it includes limited Sanskrit content, it's an excellent resource for training initial models with more common languages before transitioning to Sanskrit.

4. **Sanskrit-English Lexicon Dataset**
   Website: https://old.datahub.io/dataset/sanskrit-english-lexicon
   A curated lexicon for Sanskrit-to-English translations, focusing on individual words. It's suitable for foundational translation models and for enriching word-level understanding in sentence translation tasks.

5. **Sanskrit Word Segmentation Dataset (Zenodo)**
   DOI:https://doi.org/10.5281/zenodo.803508
   This dataset was developed to address challenges in Sanskrit word segmentation, where incorrect breaking of compound words can lead to misinterpretation. It provides properly segmented word data, critical for pre-processing in NLP models. For instance, अहं गॅच्छा (anh: gaccha)and गॅच्छामि (gacchami) mean the same thing.

6. **Bhagavad Gita Translation**
   Website: https://www.holy-bhagavad-gita.org/chapter/2/verse/1
   This source presents each verse in Sanskrit along with its IAST transliteration and English translation, allowing for multi-level input for training models (original script, transliteration, and translation).

1. Dataset Preparation Phase

Preparing a robust dataset is critical for the success of any translation model, especially when dealing with a linguistically complex language like Sanskrit.

A. **Data Collection**

Sources:

- **Parallel Corpora:**
  - Sanskrit Bible (New Testament): aligned sentence pairs in JSON format
  - Bhagavad Gita (verse-by-verse English-Sanskrit with IAST)
  - Sāmayik Corpus, AI4Bharat, OPUS, and other multilingual datasets
- **Lexicons & Word Lists:**
  - Sanskrit–English Lexicon (for individual word mappings)
  - Morphological Tagging Resources (e.g., INRIA, Morphogen)
- **Segmented Datasets:**
  - Zenodo dataset for correct compound word breaking (sandhi splitting)

B. Preprocessing Steps

1. **Normalization:**
    - Convert Sanskrit data to IAST (Latin transliteration) to simplify modeling and tokenization.
    - Ensure consistent Unicode normalization (NFKC for Devanagari, NFC for IAST).
2. **Cleaning & Filtering:**
    - Remove noisy data (e.g., incomplete pairs, misaligned segments).
    - Remove scripture references, annotations, or HTML tags.
    - Filter sentences to have similar lengths or lengths within a defined threshold (e.g., max 20–25 tokens).
3. **Segmentation:**
    - For Sanskrit: use tools like the Zenodo word segmentation corpus to split collated words accurately.
    - For English: perform lemmatization and stopword removal if needed.
4. **Tokenization:**
    - Use SentencePiece or subword tokenizers (BPE, WordPiece) for handling rare words and improving vocabulary coverage.
    - Create aligned vocabularies for English and Sanskrit (especially useful for LSTM-based models).
5. **Vectorization:**
    - Convert tokens into indices using tokenizer vocabulary.
    - Pad or truncate sequences to fixed length for batch training.
6. **Data Splitting:**
    - Standard practice:
        - **Training set:** 80%
        - **Validation set:** 10%
        - **Test set:** 10%
    - Ensure sentence pairs are not duplicated across splits.

2. Model Testing Phase

This phase involves implementing your machine translation model and conducting controlled experiments to optimize performance.

A. Model Architectures Used

1. **Baseline: Encoder–Decoder with LSTM**
    - Encoder processes input sequence (English), converts to context vector
    - Decoder generates target sequence (Sanskrit) one word at a time
2. **With Attention Mechanism** (Bahdanau/Luong)
    - Helps model focus on specific parts of the input while decoding
    - Especially useful for longer sentences and Sanskrit's free word order
3. **Statistical Machine Translation (SMT)**
    - Phrase-based SMT or alignment-based SMT with morphological features
    - POS tagging and inflectional analysis as additional features

4. **Hybrid Models**
    - Combine SMT preprocessing (e.g., word alignment) with Neural MT modeling
    - Useful for low-resource settings like Sanskrit
5. **Transformer (if explored)**
    - Self-attention allows modeling non-local dependencies better than RNNs
    - Use pretrained embeddings or multilingual models if training data is limited

B. Training Configurations

- **Hyperparameters:**
    - Batch size: 64–128
    - Epochs: 20–50 (early stopping with patience)
    - Embedding size: 256–512
    - LSTM hidden units: 512–1024
    - Dropout: 0.2–0.5
- **Optimization:**
    - Loss function: Sparse Categorical Crossentropy
    - Optimizer: Adam with learning rate decay
    - Gradient clipping to avoid exploding gradients
- **Infrastructure:**
    - Use GPUs (via TensorFlow or PyTorch with CUDA)
    - Optional: Try MXNet or HuggingFace Transformers for multi-GPU training

3. Evaluation Phase

This phase measures how well your model performs and identifies areas of improvement.

A. Quantitative Evaluation Metrics

1. **BLEU Score (Bilingual Evaluation Understudy):**
    - Measures n-gram overlap between model output and reference translation
    - Common for MT, but less effective for morphologically rich languages like Sanskrit
2. **METEOR Score:**
    - Considers stemming, synonym matching, and word order penalties
    - Better than BLEU for Sanskrit–English comparisons
3. **ChrF Score (Character F-score):**
    - Character-level metric suited for morphologically rich languages
    - Can capture similarity even when inflectional variants differ
4. **Loss Curves (Train vs Validation):**
    - Ensure that model is not overfitting or underfitting
    - Use early stopping based on validation loss

B. Qualitative Evaluation

1. **Manual Sentence Review:**

- Human annotators evaluate sentence pairs for:
  - **Fluency**: Is the output grammatical and natural?
  - **Adequacy**: Does the translation preserve the full meaning?
  - **Faithfulness**: Is the Sanskrit semantics accurately captured?

2. **Error Analysis:**
   - Identify patterns in translation errors:
     - Misaligned syntax
     - Incorrect verb inflection
     - Improper handling of compound words (sandhi errors)
     - Pronoun/reference mismatches
3. **Back-Translation Accuracy:**
   - Translate Sanskrit output back to English and compare with original
   - Helps validate how much semantic content was preserved

Special Considerations for Sanskrit

- **Sandhi and Samāsa (Compounding):**
  These often cause segmentation challenges and require specific preprocessing.
- **Free Word Order:**
  Sanskrit allows SOV, OSV, VSO, etc., so attention-based models are better than plain seq2seq.
- **Morphology-Rich Vocabulary:**
  Sanskrit uses extensive inflections for tense, mood, gender, case, and number—consider morphological analysis or tagging during preprocessing and evaluation.

Summary:

Phase	Key Focus	Tools/Techniques
Data Preparation	Quality, alignment, segmentation	Tokenization, normalization, sandhi splitting
Testing	Model design, learning dynamics	LSTM + attention, transformers, SMT
Evaluation	Translation quality, linguistic validity	BLEU, METEOR, ChrF, human feedback

Literature Survey

Amrith Krishna, Pavankumar Satuluri, and Pawan Goya, *A Dataset for Sanskrit WordSegmentation*

The authors have released a dataset which contains 80,000 sentences that have been correctly segmented to remove the inconsistencies that otherwise occur in Sanskrit to English translation as Sanskrit is a language in which words are collated and if broken down incorrectly could have opposite meanings. The authors have created the dataset to make Sanskrit more accessible to the computing field as they have undertaken the basic

preprocessing that we wouldotherwise have to undertake.

Rashmi Jha, Aman Jha, Deeptanshu Jha, and Sonika Jha, ***Is Sanskrit the most suitablelanguage for Natural Language Processing?***

This paper discusses if Sanskrit is a language fit for NLP. The authors conclude that Sanskrit is a viable language for NLP as compared to other languages its grammar and syntax is easier tounderstand. Additionally, as words like it is, and often I do not have separate words in Sanskrit but can be easily decoded due to inflections it reduces a four-letter English word to 2 letters in Sanskrit making it easier to decode and reducing storage space. Lastly, the authors found that due to the well thought out structure of the language with easy to follow grammatical rules, it makes it possible to correctly translate and decipher the language.

Lee, Young-Suk, ***Morphological Analysis for Statistical Machine Translation*** **(IBM)**

This paper discusses the use of an older technique of translation called Statistical Machine Translation but with the addition of Morphological features. These Morphological and syntactic features are also called 'part-of-speech' tags. At its core this paper takes the corpus of each languages and utilizes the part-of-speech tagging to create features for the statistical analyses and model. More specifically this paper utilizes prefix and suffix information and
part-of-speech mapping to power their model.

Priscila Aleixo, Thiago Alexandre Salgueiro Pardo, ***Finding Related Sentences in Multiple Documents for Multidocument Discourse Parsing of Brazilian Portuguese Texts***

The goal of this paper was to find some measure that allows for similar sentences to be found in multiple documents that are the same topic. After pre-processing the sentences such thatall the sentences have their stopwords removed and the whole document is lemmatized, a lexical similarity measure was applied. After all the different measures were taken, the authors looked atrecall, precision, f-measure, and cosine similarity all along with various thresholds to see which measure provided the best results (of which recall was preferred). The measures included tests such as "word overlap + stoplist" and "cosine + lemmatization". There were two main conclusions that can be drawn from the study. The first being that the best measure that provided the best results, under the assumption that recall is the better measure, was cosine similarity + lemmatization which had about a 93-100% recall around a threshold of 0.1-0.2. The second conclusion was that the results were overall better when looking at documents in Portuguese as opposed to English for Brazilian Portuguese texts.

P Bahadur, A Jain, D.S.Chauhan, *English to Sanskrit Machine Translation*

This paper describes the main differences between English and Sanskrit, and then goes onto explain a general algorithm of translating between the two. It goes into great detail of how we can use a top-down processing style, where one can start with the Sanskrit text -> break it up into tokens and phrases -> parse them -> take that and parse it into English grammar -> and finally convert that phrase into English. Finally, the paper discusses how the various parts of speech in the English language (noun, verb, adjective, adverb) is stored into a database with added tokens to the word to make recognition easier for the machine.

References:

[1]. Amrith Krishna, Pavankumar Satuluri, and Pawan Goya, *A Dataset for Sanskrit WordSegmentation*

[2]. Lee, Young-Suk, *Morphological Analysis for Statistical Machine Translation*

(IBM)P Bahadur, A Jain, D.S.Chauhan, *English to Sanskrit Machine Translation*

[3]. Priscila Aleixo, Thiago Alexandre Salgueiro Pardo, *Finding Related Sentences in Multiple Documents for Multidocument Discourse Parsing of Brazilian Portuguese Texts*

[4]. Rashmi Jha, Aman Jha, Deeptanshu Jha, and Sonika Jha, *Is Sanskrit the most suitablelanguage for Natural Language Processing?*

[5]. *https://timesofindia.indiatimes.com/city/mumbai/ai-datasets-by-iit-bombay-to-simplify-indian-texts-help-in-ai-research/articleshow/121546993.cms?utm_source=chatgpt.com*

Additional Datasets & Corpus Resources

- **IIT Bombay AIKosh Sanskrit Text Dataset**
  IIT Bombay recently released a curated dataset of ~218,000 Sanskrit sentences (≈1.5 million words), drawn from classical texts on subjects like astronomy, mathematics, and medicine, spanning over 1,800 years. It's publicly available via the AIKosh portal (github.com).
- **SAHAAYAK 2023**
  A 1.5 million-sentence parallel corpus between Sanskrit and Hindi across multiple domains (news, conversations, literature), meticulously mined and cleaned for MT tasks (arxiv.org).

- **Sāmayik: English–Sanskrit Parallel Dataset**
  Comprises ~53,000 prose sentence pairs in modern Sanskrit and English. Benchmarks show it enhances translation quality on contemporary text significantly (arxiv.org).
- **Vedavani: Vedic Sanskrit ASR Dataset**
  A 54-hour audio dataset (30,000+ samples) from Rig and Atharva Veda, aimed at speech recognition research—useful for building speech-enabled translation systems (arxiv.org).
- **CNeRG EvalSan Toolkit**
  Offers a suite of evaluation tasks (embedding evaluation, segmentation, etc.) specifically designed for Sanskrit NLP & embeddings (cnerg-iitkgp.github.io).
- **AI4Bharat IndicNLP Catalog**
  Contains a wide-ranging collection of parallel corpora, including English–Sanskrit and Sanskrit–Hindi, as well as wordnets, pre-trained embeddings, and more (github.com).
- **Valmiki Rāmāyaṇa Dataset (Reddit source)**
  An open-source corpus of 24,000+ shlokas with Sanskrit script, transliteration, English glosses, and explanations. Community clean-up is underway and contributions are welcome (reddit.com).

Additional Key Literature

1. **Pragya & Akanksha Shukla (2014)**
   English Speech to Sanskrit Speech using Rule-based Translation — Covers a pipeline for speech-to-text conversion followed by rule-based translation and speech synthesis (ijcaonline.org).
2. **Anusaaraka Machine Translation System**
   Developed by IIIT-Hyderabad, this system applies Pāṇinian grammatical rules for English→Indian language translation, and serves as a classical-to-contemporary ML system (en.wikipedia.org).
3. **Anuvaad (Hindi ↔ Sanskrit MT)**
   A rule-based Hindi-Sanskrit system achieving ~93% translation accuracy on a 110-example test set (igi-global.com).
4. **GA-based Sanskrit→Hindi MT (2018)**
   This Genetic Algorithm–based model explores evolutionary optimization for MT and provides alternative approaches to rule- or statistical-based translation (link.springer.com, en.wikipedia.org).
5. **Multilingual MT Framework with Sanskrit as Interlingua (2019)**
   Explores using Sanskrit as a pivot ("interlingua") for translating between multiple languages—leveraging Sanskrit's regular grammar structure (link.springer.com).
6. **Sanskrit MT Comparative Study (2016)**
   An overview and comparison of multiple Sanskrit MT systems, summarizing their architectures and performance metrics (ijcaonline.org).

Further Recommendations for Exploration

- **Survey on Ancient Indian Language MT (2020)**
  A broad survey covering diverse MT methods (direct translation, interlingua, statistical, rule-based) applied to ancient Indian languages (reddit.com, ijert.org).
- **Evaluation Metrics in MT**
  Reference standard metrics like BLEU, METEOR, and other quality evaluation frameworks important for rigorously benchmarking your model .