Odia-English Neural Machine Translation System
Using Fine-Tuned mT5-Small Model

Project Report

November 20, 2025

Group Members

- Abhishek Chaudhari Kurmi (123CS0949)
- Amar Jyoti Panigrahi (423MA5048)
- Anmol Nayak (123CE0098)
- Anubhav Dash (423MA5040)
- Smruti Ranjan Mohanty (423MA5046)

Abstract

This report presents the development and comprehensive evaluation of a neural machine translation system for Odia-English language pairs. The project utilizes a fine-tuned mT5-small model with Rotary Position Embeddings (RoPE) to perform translation tasks. Despite achieving structural grammatical correctness in many cases, the model's average BLEU score of 0.139 indicates significant room for improvement in translation quality. Detailed analyses of translation outputs reveal strengths in proper noun recognition and sentence structure, alongside limitations in semantic accuracy and contextual understanding. Recommendations for future enhancements include increasing training data diversity, extending training duration, and adopting larger model architectures.

Contents

1 Executive Summary

This project implements an end-to-end neural machine translation system capable of translating text from Odia (an Indian language with approximately 45 million speakers) to English using state-of-the-art transformer architecture.

1.1 Key Results

- **Average BLEU Score:** 0.139
- **Average Word Overlap:** 0.380
- **Translation Quality:** Requires significant improvement
- **Best Performing Sentence:** 0.417 BLEU score
- **Model Parameters:** ~300M (mT5-small)

2 Introduction

2.1 Motivation

Neural Machine Translation (NMT) has revolutionized language translation, but low-resource language pairs like Odia-English remain challenging. This project addresses the need for accessible translation systems for Indian languages.

2.2 Objectives

1. Develop a functional Odia-English translation system
2. Leverage pre-trained multilingual models for transfer learning
3. Implement comprehensive evaluation metrics
4. Create a modular, extensible codebase

3 System Architecture

3.1 Model Overview

The system is built on Google's mT5-small (multilingual T5) architecture with the following enhancements:

```
Input: Odia Text -> Tokenizer -> Encoder -> Decoder -> Output: English Text
                     |
              Task Prefix: "translate Odia to English: "
```

Figure 1: High-Level System Architecture

3.2 Model Configuration

Table 1: Model Architecture Parameters

Parameter	Value	Description
Model Size	mT5-small	300M parameters
Hidden Size	768	Model dimensionality
Attention Heads	12	Multi-head attention
Encoder/Decoder Layers	8	Transformer layers
Max Sequence Length	128	Input/output limit
Vocabulary Size	250,112	mT5 multilingual vocab

3.3 Training Configuration

Table 2: Training Hyperparameters

Parameter	Value	Description
Learning Rate	1×10^{-4}	Initial learning rate
Batch Size	6	Training batch size
Epochs	20	Maximum training epochs
Optimizer	AdamW	Adaptive learning rate
Scheduler	Cosine with Restarts	LR scheduling
Gradient Clipping	1.0	Gradient norm clipping
Label Smoothing	0.1	Regularization
Accumulation Steps	4	Gradient accumulation

3.4 Technical Components

3.4.1 Rotary Position Embeddings (RoPE)

The system incorporates RoPE to enhance positional awareness:

$$\text{RoPE}(x, pos) = x \cdot \cos(pos \cdot \theta) + \text{rotate\_half}(x) \cdot \sin(pos \cdot \theta)$$

where $\theta = 10000^{-2i/d}$ for dimension i out of d dimensions.

4 Implementation Details

4.1 File Structure

```
project/├──
 config.py                  # Configuration classes├──
 tokenizer.py               # Text processing├──
 model.py                   # Model architecture├──
```

```
│ dataset.py                    # Data loading├──
│ improved_kaggle_train.py      # Training script├──
│ inference.py                  # Translation engine├──
│ evaluate_translations.py      # Evaluation metrics├──
│ plot_training_metrics.py      # Visualization└──
  t5_weights_only.pt            # Trained weights
```

4.2 Key Features

Training Features:

- Comprehensive logging and checkpointing
- Real-time translation quality monitoring
- Automatic best model selection
- Memory-efficient training with gradient accumulation
- Multi-GPU support with DataParallel

Inference Features:

- Simple Python API for integration
- Command-line interface for batch processing
- Interactive translation mode
- Error handling and recovery

5 Performance Evaluation

5.1 Test Dataset

The model was evaluated on 3 representative Odia-English sentence pairs:

1. **Media/Entertainment:** Television awards ceremony viewership
2. **News/Politics:** Ukrainian-Russian conflict reporting
3. **Labor/Business:** MLB union affiliation news

5.2 Evaluation Metrics

5.2.1 BLEU Scores

BLEU (Bilingual Evaluation Understudy) measures n-gram precision between model output and reference translations.

Sentence	BLEU Score	Quality Assessment
1	0.000	Poor - Major semantic errors
2	0.000	Poor - Missing key context
3	0.417	Moderate - Acceptable quality
Average	**0.139**	**Poor Overall**

5.2.2 Word Overlap Analysis

Sentence	Word Overlap	Shared Concepts
1	0.333	Numbers, basic structure
2	0.286	Military terms, success
3	0.522	Organization names
Average	**0.380**	**Moderate Overlap**

5.2.3 N-gram Precision Analysis

Metric	Sent. 1	Sent. 2	Sent. 3	Average
1-gram Precision	0.545	0.375	0.684	0.535
2-gram Precision	0.300	0.200	0.500	0.333
3-gram Precision	0.111	0.071	0.353	0.178
4-gram Precision	0.000	0.000	0.250	0.083

5.3 Detailed Translation Analysis

5.3.1 Sentence 1: Critical Semantic Error

Input (Odia):

> ଏହି ଟେଲିଭିଜନ ପୁରସ୍କାର ସମାରୋହ 2021 କାର୍ଯ୍ୟକ୍ରମ ତୁଳନାରେ ପ୍ରାୟ 1.5 ନିୟୁତ ଦର୍ଶକଙ୍କୁ ହରାଇଥିଲା ।

Model Output:

> ”The telescope created nearly 1.5 million viewers in the 2021 program.”

Reference:

> ”The television awards ceremony lost roughly 1.5 million viewers compared to its 2021 program.”

Issues Identified:

- Critical semantic error: ”telescope” instead of ”television”
- Incorrect verb: ”created” instead of ”lost”
- Missing context: ”awards ceremony” not translated
- Structural problems in comparison understanding

5.3.2 Sentence 2: Missing Key Entity

Input (Odia):

ପୂର୍ବରେ ଋଷିଆ ସେନା ବିରୋଧରେ ୟୁକ୍ରେନୀୟ ସେନା ପକ୍ଷରୁ ଜାରି ହୋଇଥିବା ପ୍ରତିଆକ୍ରମଣରେ ନୂଆ ସଫଳତା ମିଳିଛି ।

Model Output:

"There's a new success in the Russia military against the Russian force, said the U.S. government."

Reference:

"Ukrainian forces are claiming new success in their counteroffensive against Russian forces in the east."

Issues Identified:

- Missing key entity: "Ukrainian forces" not identified
- Incorrect attribution: Added "U.S. government" (hallucination)
- Lost context: "counteroffensive" concept missing
- Geographical reference "in the east" omitted

5.3.3 Sentence 3: Best Performance

Input (Odia):

ଏମଏଲବି ଖେଳାଳି ସଂଘ ଶେଷରେ ଏଏଫଏଲ □ ସିଆଇଓର ସଦସ୍ୟ ହେବ ଏବଂ ବିଭିନ୍ନ ଶିଳ୍ପର ଅନ୍ୟ ୫୭ଟି ସଂଘ ସହିତ ଯୋଡ଼ି ହେବ ।

Model Output:

"The MLB Players Association will be a member of the AFL-CIO and another 75 other unions with other projects."

Reference:

"The MLB Players Association will finally be a member of the AFL-CIO, affiliating with 57 other unions across industries."

Strengths:

- Correct entity recognition: "MLB Players Association", "AFL-CIO"
- Proper sentence structure and grammar
- Accurate core meaning preservation

Minor Issues:

- Number discrepancy: "75" vs "57"
- Missing adverb: "finally" not translated
- Imprecise phrasing: "other projects" vs "across industries"

6 Strengths and Limitations

6.1 Technical Strengths

- Modular, well-structured codebase
- Comprehensive evaluation framework
- Real-time monitoring capabilities
- Flexible configuration system
- Multi-modal inference options

6.2 Model Strengths

- Good at proper noun recognition (organizations, numbers)
- Maintains basic sentence structure
- Handles multilingual tokenization effectively
- Reasonable performance on domain-specific content

6.3 Limitations

6.3.1 Model Limitations

- Poor average BLEU score (0.139)
- Inconsistent translation quality across samples
- Semantic errors in complex sentences
- Limited contextual understanding
- Vocabulary gaps for specialized terms
- Tendency to hallucinate information

6.3.2 Technical Limitations

- Small model size (mT5-small) limits capacity
- Limited training data diversity
- No domain adaptation mechanisms
- Basic preprocessing pipeline
- Insufficient training epochs for convergence

7 Recommendations for Improvement

7.1 Short-term Improvements

1. **Increase Training Data:** Acquire more diverse Odia-English parallel corpus
2. **Extended Training:** Train for 50+ epochs with early stopping
3. **Data Augmentation:** Implement back-translation and paraphrasing
4. **Fine-tune Hyperparameters:** Experiment with learning rates and batch sizes

7.2 Long-term Improvements

1. **Larger Model:** Upgrade to mT5-base or mT5-large (580M+ parameters)
2. **Domain Adaptation:** Train separate models for news, entertainment, technical domains
3. **Ensemble Methods:** Combine multiple models for better predictions
4. **Human Evaluation:** Conduct user studies for qualitative assessment
5. **Post-processing:** Implement rule-based corrections for common errors

8 Conclusion

This project successfully demonstrates the implementation of a neural machine translation system for Odia-English, achieving functional translations with correct grammatical structure in many cases. However, the average BLEU score of 0.139 indicates significant room for improvement, particularly in semantic accuracy and contextual understanding.

The best-performing translation (Sentence 3, BLEU: 0.417) shows that the model has learned fundamental translation capabilities, especially for proper nouns and straightforward sentence structures. The primary challenges lie in:

- Semantic disambiguation (e.g., ”telescope” vs ”television”)
- Entity recognition in complex contexts
- Maintaining factual accuracy without hallucination

With increased training data, longer training periods, and a larger model architecture, this system has the potential to achieve production-quality translations for the Odia-English language pair.

9 Code Repository Structure

The complete codebase is organized into the following modules:

```
# Configuration Management
class ImprovedConfig:
    hidden_size: int = 768
    num_hidden_layers: int = 8
    learning_rate: float = 1e-4
    # ... additional parameters

# Tokenization
class ImprovedTokenizer:
    def __init__(self, config):
        self.tokenizer = MT5Tokenizer.from_pretrained(
            config.pretrained_model_name
        )

# Model Architecture
class T5TranslationModel(nn.Module):
    def __init__(self, config):
        self.t5_model = MT5ForConditionalGeneration
            .from_pretrained(config.pretrained_model_name)
```

10 Usage Examples

10.1 Training

```
python improved_kaggle_train.py
```

10.2 Inference

```
# Example: Using the translator
from inference import OdiaTranslator

translator = OdiaTranslator("t5_weights_only.pt")

# Odia text: ⍰⍰⍰" ⍰⍰⍰⍰ " (Mu bhala achhi - I am fine)
translation = translator.translate("\u0b2e\u0b41\u0b01␣\u0b2d\u0b32␣
   \u0b05\u0b1b\u0b3f")
print(translation)  # Output: "I am fine"
```

10.3 Evaluation

```
python evaluate_translations.py
```

11 Team Contribution Summary

Name	Primary Responsibilities
Smruti Ranjan Mohanty	Model building, hyper-parameter tuning, and large-scale training.
Anubhav Dash	Training orchestration, evaluation script development, and testing automation.
Amarjyoti Panigrahi	Maintaining pull requests, documentation pipeline, and overall project report preparation.
Anmol Nayak	Data preprocessing, tokenizer design, vocabulary construction.
Abhishek Chaudhari Kurmi	Dataset curation, tokenizer enhancement, and data QA.