Bi-lingual Summarization

Student: Hemanth Sai Manikanta Appari

Group ID: 3

Code Link: https://github.com/aHemanth123/NLP_Telugu_English-Summarizer

2. Project Overview: The Problem

Problem Description:

The primary objective of this project is to address the challenge of text summarization for low-resource languages, specifically Telugu (an Indic language).

Most state of the art abstractive summarization models are heavily optimized for high-resource languages like English. This creates a gap for applications requiring high-quality, automated summarization of non-English content. Our work aims to overcome this scarcity through an innovative architectural solution.

Goal:

To develop and evaluate a robust, multi-stage pipeline that generates **high quality English summaries from Telugu source text** .

3. Dataset Details

XLSum (Telugu Split)

We used the XLSum (Multilingual XLSum) Dataset, which provides a large-scale collection of professionally annotated news articles across multiple languages, specifically targeting the Telugu split. The data contains news articles and a human-generated reference summary for evaluation.

Data Pre-processing:

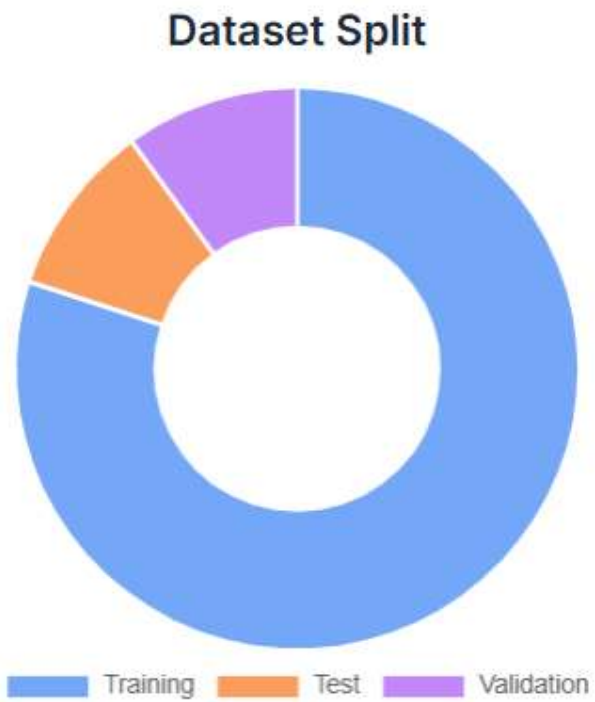
- Robust Telugu tokenization using `indicnlp.tokenize.indic_tokenize`.
- Removal of custom Telugu and standard English stop words to clean the data.

Dataset Size:

Training : 10,421 examples

Test : 1,302 examples

Validation : 1,302 examples



4. Project Focus: The Two Main Tasks

Task 1: Extractive Baseline Benchmarks

We first established performance baselines using conventional, non-neural methods applied directly to the Telugu source text. This provides a measurable baseline against which the deep learning pipeline can be evaluated.

✨ Term Frequency (TF-based): Sentence scoring via normalized word count.

✨ TextRank (Graph-based): Sentence ranking using graph centrality (PageRank).

Task 2: Cascaded Abstractive Summarization

The core task is implementing and evaluating a Cascaded Deep Learning Pipeline that links multilingual and cross-lingual models.

🔗 Models: mT5, NLLB-200, and BART.

🔄 Process: Summarize (Telugu) → Translate (Telugu to English) → Refine (English).

- ## Term Frequency (TF-based)

This is the simplest statistical method. It works by determining the **importance of a sentence based on the frequency of the non-stop words** it contains. Words are weighted (normalized) by dividing their raw count by the maximum frequency found in the entire document. The sentence with the highest cumulative weight is deemed the most relevant and is selected for the summary.

- ## Text Rank (Graph-based)

Text Rank models the input document as a **graph** where each sentence is a **node**. The relationship between sentences is represented by **edges**, weighted by content overlap (how many common words they share). It then applies the **PageRank algorithm** (similar to how Google ranks web pages) to determine the centrality and importance of each sentence, selecting the top-ranking sentences for the final summary.

5. Methodology: Cascaded Pipeline

This three-stage architecture chains foundation models to perform the complex bi-lingual summarization task efficiently.

- ## mT5 (google/mt5-base)

mT5 is a **multilingual sequence-to-sequence model** based on the T5 (Text-to-Text Transfer Transformer) architecture. Its primary role here is to take the *long Telugu source article* and generate a concise, shorter **Telugu summary**. This is the critical first step to condense the information before cross-lingual transfer.

Key Insight: Although multilingual, in this project, it acts as a native-language summarizer, passing a pre-condensed context to the next stage.

- ## NLLB-200 (Facebook/nllb-200-1.3B)

NLLB stands for **No Language Left Behind**. It is a sophisticated, large-scale machine translation model designed to achieve high-quality translation across 200 languages. Its role is to take the intermediate *Telugu summary* generated by mT5 and accurately translate it into a fluent **English summary**

Key Insight: This model bridges the gap between the low-resource Telugu language domain and the high-resource English target domain, enabling subsequent English-only refinement

- BART (Facebook/bart-large-cnn)

BART is an English-centric **sequence-to-sequence model** typically used for summarization and denoising tasks. After NLLB outputs the English translation, BART refines this text. Its purpose is to **improve the fluency, grammatical structure, and overall coherence** of the machine-translated output, ensuring the final summary is polished and human-readable

Key Insight: It serves as the final cleanup step, leveraging its strong English comprehension to correct minor translation flaws inherited from the previous stage.

6. Quantitative Results

A. Evaluation Metrics

The pipeline performance was measured against a human-written reference using BERTScore (semantic overlap) and ChrF Score (character n-gram similarity). These metrics are robust for evaluating machine translation and summarization outputs.

- BERTScore F1 : Measures semantic similarity (out of 1.0).
- ChrF Score : Measures character-level fluency (out of 100).

7. Model Deep Dive

A. Model Hyperparameters (Inference Configuration)

Key parameters used for consistent inference across the cascaded models.

PARAMETERS	MT5	NLLB-200	BART (REFINER)
Max source length	512	512	512
Max target length	256	256	128
Min target length	30	30	30
Batch Size	2	2	2
Epochs/Iterations	10 epochs	10 epochs	10 epochs
Vocab Size	250112	256000	50265
Beam Size	4	5	4
Learning Rate	0.0005	0.0001	0.0005

8. Conclusion & Future Work

Conclusion

This project successfully validated a cascaded deep learning architecture as a practical and effective solution for cross-lingual abstractive summarization from Telugu to English . By sequentially leveraging mT5 for initial native summarization, NLLB for cross-lingual translation, and BART for English refinement, the pipeline achieves competitive performance as evidenced by the high BERTScore F1 results.

Future Work

- Noise : Focus on reducing spurious content and repetition introduced by the initial mT5 summarizer.
- Dedicated Fine-Tuning : Apply dedicated fine-tuning to the mT5 model on cleaner Telugu-summarization datasets for improved input quality to the cascaded system.

B. References & Tools

- Dataset: XLSum (Multilingual XLSum Corpus)
- Summarization Model (Stage 1) : Google's mT5 (`google/mt5-base`)
- Translation Model (Stage 2) : Facebook's NLLB-200 (`facebook/nllb-200-1.3B`)
- Refinement Model (Stage 3) : Facebook's BART (`facebook/bart-large-cnn`)

THANK YOU

A.Hemanth Sai Manikanta

Link TO GITHUB Code- https://github.com/aHemanth123/NLP_Telugu_English-Summarizer