**Methodology**

Moodeng-MT: Filipino Tweet Preprocessing and Translation

August 2025

Methodology

Generated on 2025-08-26 18:24:25

# 1. Data Sources

* Primary parallel dataset: filipino\_english\_parallel\_corpus.csv (text, english\_translation or src, tgt)
* Enhanced dataset: full\_enhanced\_parallel\_corpus.csv (src/tgt; optional src\_enhanced/tgt\_enhanced)
* Tweet normalization outputs: tweets\_id\_filipino\_text\_normalized.csv; filtered: tweets\_id\_filipino\_text\_only.csv
* Auxiliary logs: logs/normalization\_log.jsonl; batch\_processing.log

# 2. Preprocessing Pipeline

We apply a Filipino-aware normalization pipeline implemented in normalizer.py with English preservation and punctuation handling. Key stages and rules:

1. Text cleaning and whitespace normalization
2. Gibberish/keyboard-smash removal (conservative)
3. Social-media artifact handling (mentions, hashtags, URLs)
4. Orthographic normalization (o↔u, e↔i, etc.) and slang expansion
5. Token split/merge, transpositions, and final formatting

Outputs preserve original terminal punctuation (?!), reducing repeats, and add a period only when none exists.

* Language filtering: Spanish confidence > 0.3 excluded; Filipino confidence > 0.1 included
* Final dataset QC: length 10–500 chars; word count 2–100; duplicate removal
* Scripts: extract\_tweet\_data.py → normalize\_csv\_tweets.py → remove\_spanish\_from\_filipino.py → final CSV

# 3. CalamanCy-enhanced Preprocessing

CalamanCy (Tagalog NLP) augments the corpus with:

* Tagalog-aware tokenization and sentence boundaries
* Linguistic complexity features (POS, dependency, morphology)
* Quality validation (grammar, entities) and optional augmentation
* Enhanced columns: src\_enhanced/tgt\_enhanced; metadata: complexity\_score, quality\_score, tagalog\_complexity, is\_augmented
* Batch process: batch\_process\_calamancy.py; resumable with intermediate enhanced\_batch\_XXX.csv

# 4. Train/Validation Split

Default 80/20 split, optionally guided by curriculum thresholds on complexity metrics.

# 5. Model and Tokenizer

Base model: facebook/mbart-large-50-many-to-many-mmt; Tokenizer: MBart50Tokenizer with src\_lang=tl\_XX, tgt\_lang=en\_XX.

# 6. Parameter-efficient Fine-tuning (LoRA)

LoRA adapters target attention/FFN modules for efficient training; best adapters saved to fine-tuned-mbart-tl2en-best/.

# 7. Curriculum and Losses

Progressive exposure from simple to complex examples using complexity thresholds. Loss mixture includes Cross-Entropy (CE), label smoothing, focal loss, and optional R-Drop.

|  |  |  |
| --- | --- | --- |
| Phase | Threshold (example) | Loss mixture |
| Simple | low complexity | CE |
| Medium | ≤ mid complexity | 0.7 CE + 0.3 Label Smoothing |
| Complex | ≤ high complexity | 0.5 CE + 0.3 Label Smoothing + 0.2 Focal |
| Mixed | all data | 0.4 CE + 0.3 LS + 0.2 Focal + 0.1 R-Drop |

# 8. Optimization and Scheduling

AdamW optimizer; cosine schedule with warmup; gradient accumulation; mixed precision on CUDA. Typical max\_seq\_len=128; beam search for eval with length/repetition controls.

# 9. Evaluation Metrics

Validation loss and BLEU (with smoothing) on a held-out set. Sentence-level BLEU is averaged as a proxy for corpus BLEU; for formal reporting, compute corpus-level BLEU on the full validation/test set.

# 10. Inference

Use translate\_with\_model.py to load the best adapter and translate Filipino text or batch CSV columns.

* Single text: python translate\_with\_model.py --text "kamusta ka?"
* Batch CSV: python translate\_with\_model.py --input\_csv file.csv --src\_col src --out\_csv out.csv
* Adapter path override: --model\_dir fine-tuned-mbart-tl2en-best

# 11. Reproducibility

* Record seeds, learning rate schedule, and LoRA hyperparameters
* Cache and version full\_enhanced\_parallel\_corpus.csv; preserve batch\_processing.log
* Environment: Python + PyTorch + Transformers + PEFT; CalamanCy/spaCy versions noted

# References

* Tang, Y., et al. (2020). Multilingual Translation with mBART-50.
* Hu, E. J., et al. (2022). LoRA: Low-Rank Adaptation of Large Language Models.
* Honnibal, M., et al. spaCy: Industrial-Strength NLP.
* CalamanCy: Tagalog NLP toolkit (GitHub project).