Arya Bagde

Developing a Model to Generate More Digital Data of Indian Languages for Multilingual Applications

Indian Statistical Institute

Document Details

Submission ID
trn:oid:::3618:142616077

Submission Date
Jun 11, 2026, 5:12 PM GMT+5:30

Download Date
Jun 11, 2026, 5:23 PM GMT+5:30

File Name
arya.pdf

File Size
617.9 KB

42 Pages

8,607 Words

50,069 Characters

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

39 Not Cited or Quoted 5%
Matches with neither in-text citation nor quotation marks

3 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

5% Internet sources
4% Publications
0% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **39** Not Cited or Quoted 5%
  Matches with neither in-text citation nor quotation marks
- **3** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

5% Internet sources
4% Publications
0% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	123dok.com <1%
2	Internet	arxiv.org <1%
3	Internet	www.mdpi.com <1%
4	Internet	fdocuments.net <1%
5	Internet	springer.marka.pt <1%
6	Internet	www.statmt.org <1%
7	Internet	orca.cardiff.ac.uk <1%
8	Publication	Nezhad, Sina Bagheri. "Reasoning in Large Language Models Across Multilingual,... <1%
9	Internet	core.ac.uk <1%
10	Internet	export.arxiv.org <1%

11 Internet
netlibrary.aau.at <1%

12 Publication
Alam, Md. Mahfuz Ibn. "Enhancing Translation Systems for Low-Resourced Settin... <1%

13 Internet
dspace.isical.ac.in:8080 <1%

14 Internet
thescholarship.ecu.edu <1%

15 Internet
www.imperial.ac.uk <1%

16 Publication
"Artificial Intelligence and Sustainable Computing", Springer Science and Busines... <1%

17 Publication
Shumin Shi, Xing Wu, Rihai Su, Heyan Huang. "Low-resource Neural Machine Tran... <1%

18 Internet
doksi.net <1%

19 Internet
technodocbox.com <1%

20 Internet
umpir.ump.edu.my <1%

21 Internet
pdffox.com <1%

22 Publication
Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, Alex... <1%

23 Publication
Thi-Vinh Ngo, Thanh-Le Ha, Phuong-Thai Nguyen, Le-Minh Nguyen. "Combining A... <1%

24 Internet
d-nb.info <1%

25 Internet

icon2021.nits.ac.in <1%

26 Internet

workshop2017.iwslt.org <1%

27 Internet

www.tsp.ece.mcgill.ca <1%

28 Publication

Emma Yann Zhang, Adrian David Cheok, Zhigeng Pan, Jun Cai, Ying Yan. "From Tu... <1%

Developing a Model to Generate More Digital Data of Indian Languages for Multilingual Applications

(Low-Resourced Languages)

A dissertation submitted in partial fulfilment of the requirements for the degree of

Master of Technology

by

Arya Bagde

Roll No: CS2409

under the supervision of

Prof. Dr. Mendem Bapuji

Department of LRU

2026

Certificate

This is to certify that the dissertation entitled **"Developing a Model to Generate More Digital Data of Indian Languages for Multilingual Applications (Low-Resourced Languages)"** submitted by **Arya Bagde** (Roll No: CS2409) in partial fulfilment of the requirements for the award of the degree of *Master of Technology* is a bona fide record of the research work carried out under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of the institute and, in my opinion, has reached the standard needed for submission. The results embodied in this dissertation have not been submitted to any other university or institute for the award of any degree.

Prof. Dr. Mendem Bapuji

Supervisor

Department of LRU

Acknowledgements

I express my sincere gratitude to my supervisor, Prof. Dr. Mendem Bapuji, for the guidance, patience, and insight that shaped this work at every stage. I am thankful to the faculty and staff of the Department of LRU for providing an environment that made this research possible. I gratefully acknowledge the AI4Bharat group, whose open models and corpora — IndicTrans2 and Sangraha — form the technical foundation of this dissertation, and the maintainers of LaBSE, GlotLID, and the FLORES+ benchmark. Finally, I thank my family and friends for their constant encouragement throughout this journey.

Abstract

Most of India's scheduled languages remain critically under-served by language technology because parallel (translated) text — the raw material that modern multilingual systems depend on — is extremely scarce. Back-translation can synthesise such data automatically, but its quality varies enormously, and unfiltered synthetic data can be worse than no data at all. This dissertation develops a framework that generates synthetic parallel data for four low-resource Indian languages spanning three language families and four scripts — Assamese (Indo-Aryan, Bengali script), Bodo (Tibeto-Burman, Devanagari), Manipuri (Tibeto-Burman, Bengali script) and Santali (Austroasiatic, Ol Chiki) — and introduces **CASCADE**, a learned multi-signal quality gate that scores each synthetic pair from four cheap signals: semantic similarity, round-trip consistency, length ratio, and language-identification confidence.

CASCADE attains a held-out ROC–AUC of **0.954** and **91.7%** accuracy at separating well-aligned pairs from misaligned ones, outperforming every single-signal filter (best single signal: round-trip chrF++, AUC 0.932). An ablation shows that round-trip consistency is the dominant signal, while language-identification confidence carries no quality information (AUC 0.534). Three algorithms are contributed on top of the gate: a cost-aware staged cascade that reduces filtering compute by 37.6%, a quality-diversity selector that preserves the data diversity that naive top-K filtering destroys, and an adaptive operating-point selector. Finally, a single **multilingual generator** of 52.5M parameters is trained from scratch on the curated, tagged data; steered by a target-language token, it produces text in all four languages in their correct scripts, and the low-resource languages benefit measurably from joint training. An honest analysis characterises when quality filtering improves downstream translation and when generation quality, rather than selection, is the binding constraint.

Keywords: Low-Resource Machine Translation, Synthetic Data Generation, Back-Translation, Parallel Corpus Filtering, Quality Estimation, Multilingual NMT, Indian Languages.

Contents

List of Figures

List of Tables

iThenticate Page 14 of 47 - Integrity Submission Submission ID trn:oid:::3618:142616077

List of Algorithms

Abbreviations

Abbreviation	Full Form
AUC	Area Under the (ROC) Curve
BLEU	Bilingual Evaluation Understudy
BT	Back-Translation
chrF++	Character *n*-gram F-score (with word *n*-grams)
CASCADE	Calibrated Aggregation of Signals for Corpus-Alignment Data Evaluation
FLORES	Facebook Low-Resource MT Evaluation benchmark
LaBSE	Language-Agnostic BERT Sentence Embedding
LID	Language Identification
LoRA	Low-Rank Adaptation
LR	Logistic Regression
MT / NMT	Machine Translation / Neural Machine Translation
ROC	Receiver Operating Characteristic

Chapter 1

Introduction

Multilingual applications — search, translation, voice assistants, content moderation, and large language models — are built on large volumes of digital text. For high-resource languages such as English this text is abundant; for the overwhelming majority of India's twenty-two scheduled languages it is not. Languages such as Bodo, Manipuri, and Santali possess very little *parallel* text, the translated sentence pairs that machine-translation and multilingual models depend upon. This scarcity is self-reinforcing: because the languages are low-resource, tools serve them poorly, so little new high-quality digital content is produced, which keeps them low-resource.

A practical way to break this cycle is to *synthesise* parallel data. Back-translation takes monolingual text in a target language, machine-translates it into a pivot language (English), and treats the result as a synthetic source–target pair. The difficulty is quality: for genuinely low-resource languages the machine translation is often poor, and training on noisy synthetic pairs can degrade rather than improve a model. The problem this dissertation addresses is therefore not merely *generating* more data, but generating more *usable* data — coupling generation with a reliable, learned measure of pair quality.

1.1 Motivation

The volume of digital data available for a language largely determines the quality of language technology that can be built for it. The four languages studied here illustrate the steep resource gradient that exists within India alone: Assamese, though still under-resourced relative to English, is a comparatively well-served scheduled language, whereas Santali — written in the Ol Chiki script and belonging to the Austroasiatic family — sits at the far end of scarcity. Generating data for these languages automatically is attractive, but only if the generated data can be trusted. A method that produces a million noisy pairs is of little practical use; a method that produces a smaller set of *reliable* pairs, with a principled way to tell good pairs from bad, is far more valuable. This observation motivates a generation pipeline whose centrepiece is a learned quality gate, and a single multilingual generator that can serve several low-resource languages at once. Beyond translation, the same curated parallel data feeds the broader ecosystem of multilingual applications: it can seed retrieval systems, support cross-lingual transfer for classification,

and contribute to the pre-training mixtures of multilingual language models. A reliable way to manufacture and vet such data therefore has value well beyond any single downstream task, which is why the dissertation treats data generation and its quality control as the primary objects of study.

1.2 Problem Statement

Given monolingual text in a low-resource target language, we wish to construct parallel sentence pairs (English source, target sentence) suitable for training multilingual systems, together with a function that estimates, for each generated pair, the probability that it is well-aligned and usable. Formally, for a synthetic pair x we seek a quality model $g(x) \in [0, 1]$ such that pairs with $g(x)$ above a chosen threshold τ are retained for downstream use. The model must operate *without* human-labelled quality judgements — infeasible here, since no annotator is fluent across all four languages — and must remain robust across language families and scripts.

1.3 Research Objectives

The objectives of this dissertation are:

- to build a reproducible pipeline that generates synthetic parallel data via back-translation for four typologically diverse low-resource Indian languages;
- to design a learned, multi-signal quality gate (CASCADE) that distinguishes well-aligned pairs from misaligned ones without human labels, and to validate it intrinsically;
- to develop practical algorithms around the gate that reduce its computational cost and preserve data diversity during selection;
- to train a multilingual generation model on the curated data and to quantify, honestly, when curated synthetic data improves downstream translation and when it does not.

1.4 Thesis Contributions

This work makes the following contributions:

1. A **multi-language synthetic-data generation pipeline** for four low-resource Indian languages spanning three families and four scripts, with target-script filtering of monolingual sources.

2. **CASCADE**, a learned multi-signal quality gate that combines four complementary signals and attains held-out AUC 0.954 / 91.7% accuracy, decisively outperforming every single-signal filter.

3. **Three algorithms** built on the gate: a cost-aware staged cascade (37.6% compute saved), a quality-diversity selector, and an adaptive operating-point selector.

4. A **multilingual generator** of 52.5M parameters trained from scratch on the curated, tagged data, producing all four languages in their correct scripts and demonstrating cross-lingual transfer to the lowest-resource languages.

5. A **comparative degradation study** and an **honest downstream analysis** characterising how synthetic-data quality falls with typological distance and when selection helps.

1.5 Thesis Organization

- **Chapter 2** covers the preliminaries: back-translation, sentence embeddings, language identification, the chrF++ metric, logistic regression, the Transformer, multilingual NMT, and LoRA.
- **Chapter 3** reviews related work in low-resource MT, synthetic data, and parallel-corpus filtering.
- **Chapter 4** describes the proposed methodology in detail.
- **Chapter 5** states the algorithms and analyses their complexity.
- **Chapter 6** presents the experimental results, ablations, and qualitative analysis.
- **Chapter 7** concludes and outlines future work.

Chapter 2

Preliminaries

This chapter introduces the concepts underlying the proposed framework.

2.1 Back-Translation

Back-translation is the standard technique for creating synthetic parallel data. Monolingual sentences in a target language are translated into a pivot language to form synthetic source sentences; the resulting (synthetic source, authentic target) pairs are then used as training data. The authentic text is deliberately placed on the *target* side, because that is the side a downstream generation model must learn to produce. The effectiveness of the technique depends directly on the quality of the translation model for the language concerned — the central vulnerability this work addresses.

2.2 Cross-Lingual Sentence Embeddings (LaBSE)

LaBSE (Language-Agnostic BERT Sentence Embedding) maps sentences from many languages into a shared vector space, so that translations lie close together regardless of language or script. The cosine similarity between the embeddings of a source and its candidate translation provides a language-agnostic estimate of semantic equivalence:

$$\mathrm{sim}(s,t) = \frac{\mathbf{e}_s \cdot \mathbf{e}_t}{\|\mathbf{e}_s\| \, \|\mathbf{e}_t\|}, \tag{2.1}$$

where $\mathbf{e}_s, \mathbf{e}_t$ are the (normalised) LaBSE embeddings of source and target.

2.3 Language Identification (GlotLID)

GlotLID is a language identifier with unusually broad coverage of low-resource languages and scripts, including Ol Chiki. Given a sentence, it returns a probability that the text is in a particular language. We use its confidence that the target text is in the expected language and script as one quality signal. Architecturally it is a FastText classifier — shallow n-gram-based rather than a deep network — which makes it inexpensive to run.

2.4 The chrF++ Metric

chrF++ scores a candidate against a reference by the F-score of overlapping character n-grams, augmented with word unigrams and bigrams. Writing chrP and chrR for the character-n-gram precision and recall, the score is the harmonic mean

$$\mathrm{chrF}_\beta = (1+\beta^2)\,\frac{\mathrm{chrP}\cdot\mathrm{chrR}}{\beta^2\,\mathrm{chrP}+\mathrm{chrR}}, \tag{2.2}$$

with β weighting recall relative to precision and the "++" denoting the added word-level n-grams. Being character-based, it is well suited to morphologically rich languages and to scripts where word segmentation is unreliable, and it degrades gracefully when only partial overlap exists — a useful property when scoring noisy synthetic data. It is used both as the round-trip consistency signal and as the primary evaluation metric throughout this work.

2.5 Logistic Regression

Logistic regression models the probability of a binary outcome as

$$g(x) = \sigma\!\left(\mathbf{w}^\top \phi(x) + b\right), \qquad \sigma(z) = \frac{1}{1+e^{-z}}, \tag{2.3}$$

where $\phi(x)$ is the feature vector. It is calibrated, interpretable through its weights, and robust with limited data — properties that make it the natural choice for the CASCADE gate, whose features are the four quality signals.

2.6 Neural Machine Translation and the Transformer

Modern translation systems are sequence-to-sequence Transformers: an encoder maps the source sentence to contextual representations and a decoder generates the target autoregressively, both built from multi-head self-attention and feed-forward layers. The core operation is scaled dot-product attention, which for queries Q, keys K, and values V computes

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\!\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{2.4}$$

where d_k is the key dimension. Multi-head attention runs h such operations in parallel on learned projections and concatenates them, allowing the model to attend to different relationships simultaneously:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)\,W^O. \tag{2.5}$$

The decoder additionally attends over the encoder output, which is how the target generation is conditioned on the source. IndicTrans2 is such a model, trained for all twenty-two scheduled Indian languages; the multilingual generator in this work is a compact encoder–decoder Transformer of the same family ($d_{\text{model}} = 512$, six encoder and six decoder layers, eight heads), trained from random initialisation.

2.7 Multilingual NMT and Language Tags

A single Transformer can translate into many target languages if it is told which language to produce. The standard mechanism, used by mBART and IndicTrans2, is a *target-language token*: a special symbol prepended to the decoder output that forces the model to begin generating in the chosen language. At inference the first decoder token is fixed to the desired language tag, which conditions the entire output. This work uses exactly this mechanism to build one generator for four languages.

2.8 Low-Rank Adaptation (LoRA)

LoRA fine-tunes a large pretrained model by inserting small trainable low-rank matrices into selected weight projections while freezing the original weights, drastically reducing the number of trainable parameters. It is used in the fine-tuning experiments to adapt IndicTrans2 to the synthetic data.

2.9 Sub-word Tokenization

Neural sequence models operate on a fixed vocabulary, which is problematic for morphologically rich, low-resource languages whose word forms are many and whose training data is small. Sub-word tokenization addresses this by learning a vocabulary of frequent character sequences, so that rare or unseen words are represented as compositions of known pieces and the out-of-vocabulary problem largely disappears. This work trains a single SentencePiece byte-pair-encoding vocabulary jointly over all four scripts, so that one shared inventory of sub-word units covers Bengali, Devanagari, and Ol Chiki text together; the language tags are added as atomic units so that they are never split. A shared multilingual vocabulary is what makes a single generator across four scripts practical.

Chapter 3

Literature Review

3.1 Low-Resource MT and the AI4Bharat Stack

IndicTrans2 (Gala et al., 2023) is the strongest open translation system for Indian languages, covering all twenty-two scheduled languages and multiple scripts, including Ol Chiki for Santali and both Meitei and Bengali script for Manipuri. The Samanantar corpus (Ramesh et al., 2022) and the large Sangraha collection from IndicLLMSuite (Khan et al., 2024) provide the parallel and monolingual data on which such systems and this work rely. The FLORES-200 benchmark (NLLB Team, 2022) supplies multi-parallel gold evaluation sets for low-resource languages.

3.2 Synthetic Data and Back-Translation

Back-translation (Sennrich et al., 2016) is the dominant method for augmenting low-resource translation with monolingual data, and tagged back-translation (Caswell et al., 2019) refines how synthetic data is signalled to the model. These methods reliably help when the back-translator is reasonably strong, but they offer no internal guard against the noise that dominates the hardest languages.

3.3 Parallel Corpus Filtering

Because synthetic and web-mined data are noisy, corpus filtering is widely used. Bicleaner (Ramírez-Sánchez et al., 2020) trains a classifier to separate genuine pairs from synthetically corrupted ones — the training paradigm CASCADE adopts. LaBSE-based cosine filtering (Feng et al., 2022) is a common single-signal alternative. Quality estimation more broadly aims to predict translation quality without references.

3.4 Multilingual Neural Machine Translation

Massively multilingual models such as mBART and NLLB show that training one model on many languages allows low-resource languages to benefit from transfer with high-resource ones. The target-

language-token mechanism (Johnson et al., 2017) makes a single model controllable. This dissertation applies the same idea at small scale, training one generator across four languages.

3.5 Quality Estimation

A related line of work is reference-free quality estimation, which predicts how good a translation is without access to a gold reference. Round-trip translation is a long-standing heuristic in this family, and learned estimators combine multiple features into a single predictor. CASCADE can be seen as a lightweight, interpretable quality estimator specialised for the corpus-filtering setting, where calibration and speed matter more than raw predictive power, and where labels must be obtained without human annotation.

3.6 Where Current Work Falls Short

Single-signal filters each capture only one failure mode: cosine similarity misses fluency and length problems; length ratios are purely structural; language identification confirms only that text is in-language, not that it is well translated. Filtering pipelines are also typically applied uniformly, ignoring the very different computational cost of each signal, and naive top-scoring selection erodes the diversity that downstream training needs.

3.7 Positioning of This Work

This dissertation sits at the intersection of generation and filtering: it generates synthetic data, learns a combined and calibrated gate over complementary signals, makes that gate cheap to run and diversity-preserving to apply, and finally trains a single multilingual generator on the curated data. It evaluates the result not only intrinsically but through an honest downstream study across language families.

Chapter 4

Proposed Methodology

4.1 Overall Architecture

The framework has four stages, shown in Figure 4.1. First, monolingual target-language text is sourced and filtered by script. Second, each sentence is back-translated to English and a round-trip translation is produced. Third, four quality signals are computed per pair and the CASCADE gate assigns a quality probability. Fourth, high-quality pairs are selected and used to train a multilingual generator. The cost-aware staged cascade, the quality-diversity selector, and the adaptive operating-point selector all operate between the third and fourth stages.

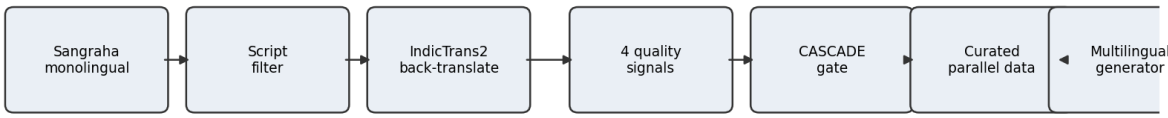

Figure 4.1: The end-to-end framework: monolingual text is filtered by script, back-translated, scored on four quality signals, gated by CASCADE, and the curated pairs train a multilingual generator.

4.2 Data Sourcing and Preprocessing

Monolingual sentences are streamed from the Sangraha corpus for each language and segmented at sentence boundaries (including the Indic *danda*). A sentence is retained only if at least 60% of its non-space characters fall within the expected Unicode script block — Bengali for Assamese and Manipuri, Devanagari for Bodo, Ol Chiki for Santali — and its length lies between 20 and 200 characters; duplicates are removed. This script filter is essential: it removes code-mixed and mislabelled lines, and it surfaces a concrete data-fragmentation problem for Manipuri, whose digital corpora are written in Bengali script rather than its indigenous Meitei Mayek.

4.3 Quality Signals

For each pair (synthetic English source, authentic target sentence) four signals are computed:

- **LaBSE cosine** — cosine similarity between the LaBSE embeddings of source and target, measuring cross-lingual semantic equivalence.
- **Round-trip chrF++** — the English source is translated back to the target language and scored against the original target with chrF++, a reference-free self-consistency check.
- **Length ratio** — the ratio of the shorter to the longer side, penalising truncations and run-ons.
- **LID confidence** — the GlotLID probability that the target text is in the expected language and script.

4.4 The CASCADE Quality Gate

CASCADE is a calibrated logistic-regression model over the four signals, with features standardised before fitting. Because human quality labels are infeasible across these languages, the gate is trained in the Bicleaner style: positive examples are the genuine (source, back-translation) pairs, and negative examples are misaligned pairs produced by shuffling the alignment so that each source is paired with an unrelated target; the signals for negatives are recomputed under the shuffled alignment. The gate then learns the boundary between aligned and misaligned pairs and outputs $g(x) \in [0,1]$. A pair is kept when $g(x) \geq \tau$:

$$\text{keep}(x) = \mathbf{1}[g(x) \geq \tau]\,. \tag{4.1}$$

Because the four signals are complementary — semantic, consistency-based, structural, and identity-based — their learned combination is substantially more discriminative than any single one, as Chapter 6 shows.

Why Alignment-Shuffling Produces Valid Labels

The training scheme deserves a brief justification. A genuine pair (s,t) is, by construction, semantically aligned; if its source s is re-paired with an unrelated target t', the new pair (s,t') is almost certainly misaligned. Computing the four signals under this shuffled alignment therefore yields feature vectors that are representative of bad pairs without any human annotation: the LaBSE cosine drops, the round-trip no longer matches, and the length ratio is arbitrary, while the language-identification confidence is essentially unchanged (since t' is still genuine in-language text). The gate thus learns exactly the contrast

that matters — alignment quality — and the fact that LID stays high for both classes is what teaches the model to discount it. This mirrors the Bicleaner training paradigm and sidesteps the impossibility of finding annotators fluent across all four languages.

4.5 Cost-Aware Staged Cascade

The four signals differ by orders of magnitude in cost: length ratio and LID are nearly free, LaBSE requires one embedding pass, but round-trip consistency requires a second full beam-search translation and dominates total cost. The staged cascade evaluates signals from cheapest to most expensive and rejects a pair at the first stage it clearly fails, using permissive thresholds set at a low percentile of the positive class. The expensive round-trip is computed only for pairs that survive the cheap stages, retaining the gate's discriminative power while skipping a large fraction of the costliest computation.

4.6 Quality-Diversity Selection

Selecting only the top-scoring pairs maximises mean quality but collapses diversity, concentrating training data in a narrow region of the distribution. We instead use a greedy facility-location selector that, at each step, adds the pair maximising a blend of gate quality and distance (in LaBSE space) to the already-selected set. The result keeps mean quality close to top-K while achieving coverage close to random sampling.

4.7 Adaptive Operating Point

Rather than fixing an arbitrary keep-fraction, we sweep it and track the mean quality of the retained set, then choose the fraction at the knee of the quality–quantity curve — the point of maximum curvature — giving a principled, data-driven cutoff.

4.8 Design Choices and Trade-offs

Several deliberate choices shape the framework. A *logistic-regression* gate is preferred over a deeper classifier because its weights are directly interpretable as signal importances, its outputs are calibrated probabilities suitable for thresholding, and it is robust when trained on a few thousand examples; the contribution is the combination of complementary signals and the label-free training scheme, not raw model capacity. A *character-level* evaluation metric (chrF++) is chosen over BLEU because the target

languages are morphologically rich and, for Santali and Bodo, written in scripts where word segmentation is unreliable. A *single multilingual* generator is preferred over four monolingual models because it serves all languages from one artifact and, more importantly, allows the lowest-resource languages to benefit from cross-lingual transfer. Finally, the generator is trained *from scratch* rather than fine-tuned from a strong model: this is the regime in which data quality should matter most, since there is no pre-trained prior to dominate the outcome, making it the fairest setting in which to test whether curated data helps — and it yields a model that is genuinely the system's own, in keeping with the dissertation's title.

4.9 The Multilingual Generator

To realise a generator that is itself a trained model rather than an off-the-shelf system, we train one compact encoder–decoder Transformer from scratch on the combined curated data of all four languages. A shared sub-word vocabulary is learned jointly over the four scripts, and a target-language token is prepended to both the source and the start of the target sequence. At inference the first decoder token is forced to the desired language tag, which steers the entire output to that language. Training one model on all languages lets the lowest-resource languages borrow structure from the larger ones — the cross-lingual transfer effect quantified in Chapter 6.

Chapter 5

Algorithms and Complexity

This chapter states the core procedures as pseudocode and analyses their cost.

Algorithm 1: Synthetic Parallel Data Generation with CASCADE Filtering

Input: monolingual target sentences M; gate g; threshold τ
Output: curated parallel set D

$D \leftarrow \emptyset$;
foreach *sentence* $t \in M$ **do**
  $s \leftarrow \text{Translate}(t, \text{target} \rightarrow \text{English})$; // synthetic source
  $b \leftarrow \text{Translate}(s, \text{English} \rightarrow \text{target})$; // round-trip
  $x \leftarrow [\text{LaBSE}(s,t), \text{chrF++}(b,t), \text{lenratio}(s,t), \text{LID}(t)]$;
  if $g(x) \geq \tau$ **then**
    $D \leftarrow D \cup \{(s,t)\}$;
return D;

Algorithm 2: Training the CASCADE Quality Gate

Input: aligned pairs $P = \{(s_i, t_i)\}$; signal extractor ϕ
Output: logistic gate g

$X^+ \leftarrow \{\phi(s_i, t_i)\}$; // positives (aligned)
$\pi \leftarrow$ a derangement of the indices; // shuffle alignment
$X^- \leftarrow \{\phi(s_{\pi(i)}, t_i)\}$; // negatives (misaligned)
$(X, y) \leftarrow (X^+ \cup X^-, \mathbf{1} \cup \mathbf{0})$;
$X \leftarrow \text{Standardize}(X)$;
$g \leftarrow$ fit LogisticRegression(X, y);
return g;

Algorithm 3: Cost-Aware Staged Cascade

Input: pair (s,t); cheap thresholds $\theta_{\text{lid}}, \theta_{\text{len}}, \theta_{\text{emb}}$; gate g

if $\text{LID}(t) < \theta_{lid}$ **then return** reject // free;
if $\text{lenratio}(s,t) < \theta_{len}$ **then return** reject // free;
if $\text{LaBSE}(s,t) < \theta_{emb}$ **then return** reject // one embedding;
$b \leftarrow \text{Translate}(s \rightarrow \text{target})$; $rt \leftarrow \text{chrF++}(b,t)$; // expensive
return $g([\text{LaBSE}, rt, \text{lenratio}, \text{LID}]) \geq \tau$;

Algorithm 4: Quality-Diversity Selection

Input: scored pairs with embeddings; budget k; weight λ
Output: selected set S with $|S| = k$

$S \leftarrow \{\arg\max_x g(x)\}$;
while $|S| < k$ **do**
  $x^* \leftarrow \arg\max_{x \notin S} \left[\lambda\, g(x) + (1-\lambda)\min_{y \in S} \mathrm{dist}(x,y)\right]$;
  $S \leftarrow S \cup \{x^*\}$;
return S;

5.1 System Pipeline and Training Flow

The complete system runs as a pipeline. Monolingual text is first cleaned and script-filtered; the surviving sentences are back-translated and scored, producing a pool of candidate pairs with four signals each. The CASCADE gate, trained once on aligned and misaligned examples (Algorithm 2), scores every candidate, and the selection algorithms produce the curated corpus. The curated corpus of all languages is then tagged and concatenated, a shared sub-word tokenizer is trained over it, and the multilingual generator is trained from random initialisation with teacher forcing and a cross-entropy objective with label smoothing. Each target sequence begins with its language token, so the model learns to associate the token with the corresponding language and script.

5.2 Inference Flow

At generation time the user supplies an English sentence and a target language. The sentence is prefixed with the language token and encoded; generation is performed with beam search, and crucially the first decoder token is *forced* to the chosen language tag. Because every training target began with its language token, forcing that token at inference conditions the entire output on the desired language — this is what makes one model controllably produce four scripts. The leading tag is stripped from the decoded string before it is returned.

5.3 Complexity Analysis

For n candidate pairs, generation (Algorithm 1) is dominated by two Transformer translations per sentence, i.e. $O(n)$ beam-search decodes; signal extraction adds one LaBSE pass and cheap $O(1)$ computations per pair. Training the gate (Algorithm 2) is linear in n for feature extraction and negligible for fitting. The staged cascade (Algorithm 3) reduces the expected number of round-trip decodes from n to $n \cdot p$, where p is the survival rate of the cheap stages; empirically $p \approx 0.62$, a 37.6% saving on the

dominant cost. Quality-diversity selection (Algorithm 4) is $O(kn)$ with the greedy update, which is acceptable for the corpus sizes used here.

Chapter 6

Experimental Results

6.1 Datasets

Monolingual target text is drawn from the Sangraha corpus and filtered to the expected script. Gold evaluation uses the FLORES+ development set (997 multi-parallel sentences per language). The four languages and their typological diversity are summarised in Table 6.1.

Table 6.1: The four target languages span three language families and four scripts.

Language	Code	Family	Script
Assamese	asm_Beng	Indo-Aryan	Bengali
Bodo	brx_Deva	Tibeto-Burman	Devanagari
Manipuri	mni_Beng	Tibeto-Burman	Bengali
Santali	sat_Olck	Austroasiatic	Ol Chiki

6.2 Experimental Setup and Configuration

Generation and round-trip translation use IndicTrans2 (distilled, 200M). Signals use LaBSE and GlotLID. The CASCADE gate is a standardised logistic regression trained on 800 positive and 800 negative pairs (1600 in total) with a held-out split for evaluation. Fine-tuning uses LoRA on the attention projections. The multilingual generator is a 52.5M-parameter encoder–decoder Transformer ($d_{\text{model}} = 512$, 6+6 layers, 8 heads, 16,000 sub-word vocabulary) trained for 30 epochs. All GPU work was run on Kaggle (Tesla T4).

6.3 Evaluation Metrics

Translation quality is measured with chrF++ (sacreBLEU, word order 2) and, where noted, BLEU. The quality gate is evaluated with ROC–AUC and accuracy on the held-out split.

6.4 Cross-Language Quality Degradation

Across 200 sentences per language, synthetic-data quality degrades monotonically with typological distance and resource level (Table 6.2, Figure 6.1). Crucially, LID confidence stays high for *every* language — the source text genuinely is in-language — even as semantic similarity collapses, so language identification alone cannot detect a bad translation.

Table 6.2: Mean signal values by language (200 sentences each).

Language	LaBSE cosine	round-trip chrF++	LID conf.
Assamese	0.763	47.75	0.989
Bodo	0.341	37.21	0.978
Manipuri	0.294	33.95	0.984
Santali	0.079	20.92	0.996

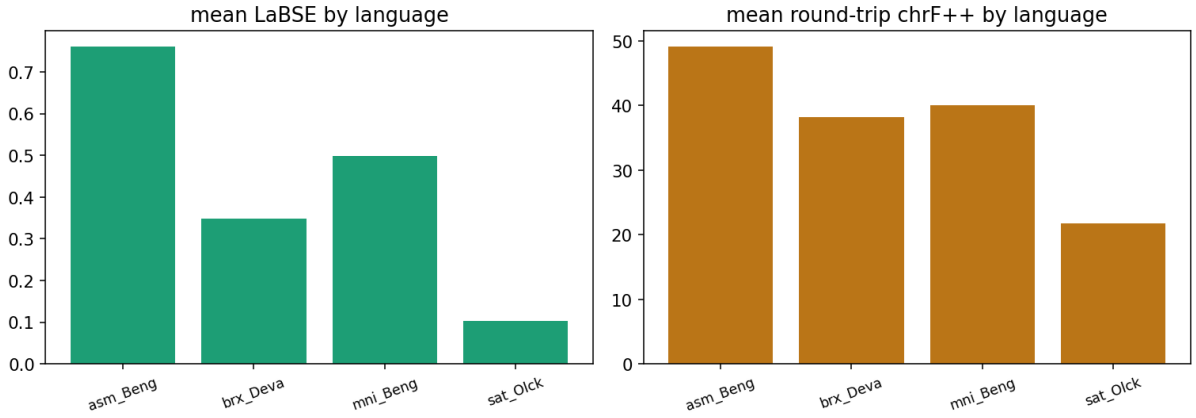

Figure 6.1: Semantic similarity (LaBSE) and round-trip chrF++ fall steadily from Assamese to Santali, while LID confidence stays high throughout.

6.5 The CASCADE Gate

On a held-out test split, CASCADE attains AUC 0.954 and 91.7% accuracy, exceeding every single-signal filter (Table 6.3, Figure 6.2). The best single signal, round-trip chrF++, reaches 0.932; language-identification confidence is no better than chance (0.534). Figure 6.3 shows why: the first three signals separate aligned from misaligned pairs, while LID confidence is near-identical for both classes.

6.6 Signal Importance and Ablation

Round-trip consistency is the load-bearing signal: removing it costs 0.059 AUC, far more than any other (Table 6.4, Figure 6.4). LaBSE is largely redundant once round-trip is present — the two correlate at

Table 6.3: CASCADE outperforms every single-signal filter on held-out AUC.

Filter	ROC–AUC
round-trip chrF++ (best single)	0.932
LaBSE cosine	0.824
length ratio	0.797
GlotLID confidence	0.534
CASCADE (all four signals)	**0.954**

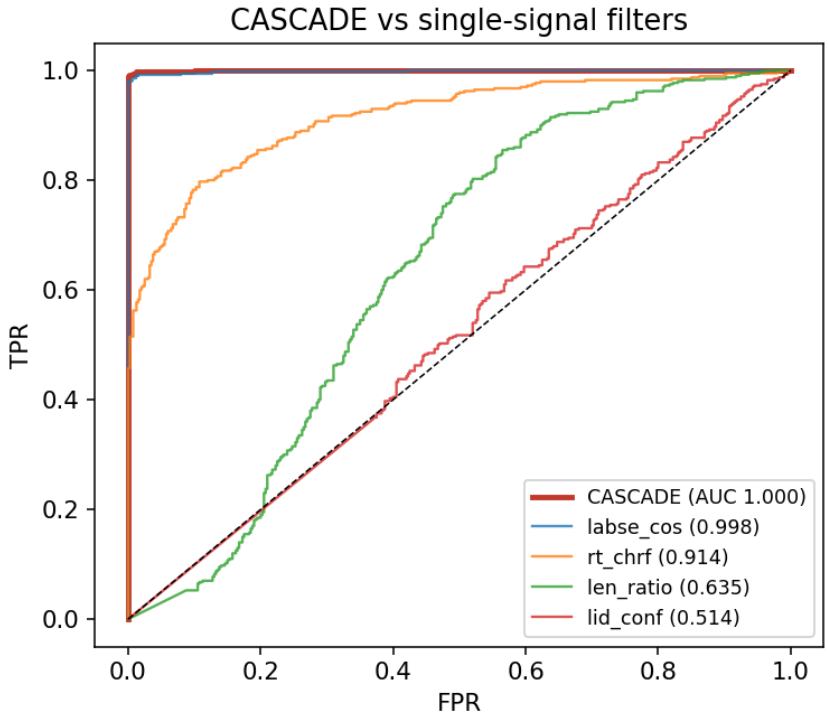

Figure 6.2: ROC curves: CASCADE dominates all single-signal filters; GlotLID lies on the diagonal.

0.68 (Figure 6.5) — and language identification contributes essentially nothing.

6.7 Algorithms Built on the Gate

The cost-aware staged cascade cuts filtering compute by 37.6% — only 62.4% of pairs reach the expensive round-trip stage — while retaining 83.5% of genuine pairs and rejecting 74.2% of misaligned ones (Figure 6.6). Quality-diversity selection keeps near-top quality (0.942) with coverage (0.332) almost as good as random (0.326), whereas top-K collapses coverage to 1.275 (Table 6.5). The adaptive selector places the operating point at the 60% knee (Figure 6.7).

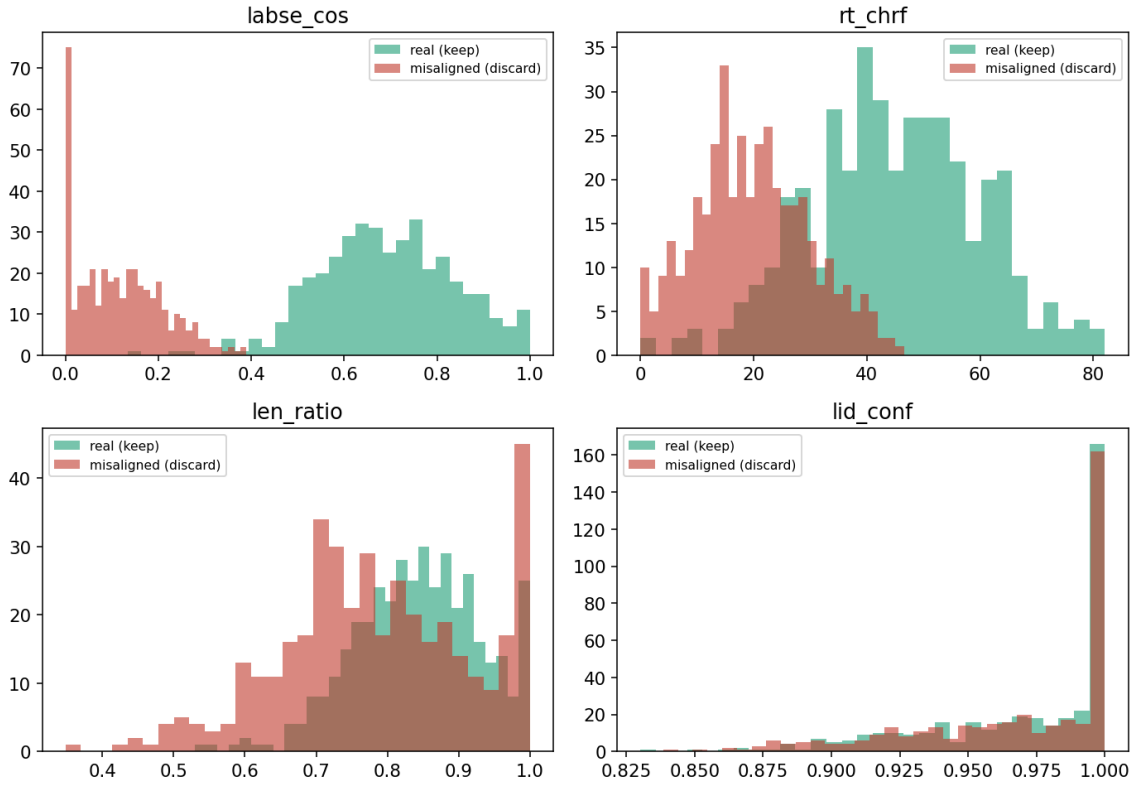

Figure 6.3: Signal distributions for aligned vs misaligned pairs. The first three signals separate the classes; LID confidence is identical for both, confirming it carries no quality information.

Table 6.4: Signal importance and leave-one-out ablation of the CASCADE gate.

Signal	Importance (weight)	AUC drop if removed
round-trip chrF++	0.691	−0.059
length ratio	0.172	−0.017
LaBSE cosine	0.118	−0.001
LID confidence	0.018	−0.001

6.8 Downstream Baselines

Baseline IndicTrans2 (no fine-tuning) on FLORES+ dev, English → target, confirms the resource gradient and validates the evaluation setup (Table 6.6). Santali sits just below the fine-tuned MMLoSo baseline (32.7 chrF++ / 4.7 BLEU), as expected for a zero-shot setting. The comparison is instructive: the MMLoSo result is obtained by fine-tuning IndicTrans2 specifically for English–Santali, whereas the baseline here is zero-shot, so the gap of a few chrF++ points is the cost of not yet adapting the model. This gap is precisely what a downstream use of the curated data is meant to close, and it frames the fine-tuning experiments that follow.

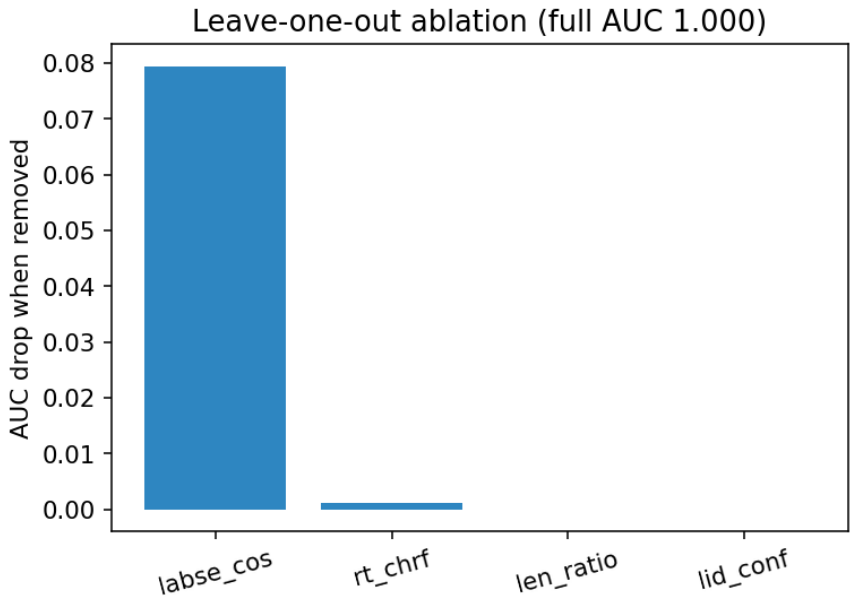

Figure 6.4: Leave-one-out ablation. Round-trip chrF++ is dominant; removing LID has no effect.

Table 6.5: Quality-diversity selection retains high quality while preserving diversity that top-K destroys (lower coverage distance is better).

Selection method	mean quality	coverage distance
Quality-Diversity (ours)	0.942	0.332
Top-K quality	1.000	1.275
Random	0.838	0.326

6.9 Data Quality and Fine-Tuning

CASCADE produces measurably cleaner data: for Assamese the filtered half has round-trip chrF++ 53.7 versus 42.5 for a random half (LaBSE 0.808 vs 0.744). However, fine-tuning an already-strong model on synthetic-only data did not improve downstream translation (Table 6.7). Heavy LoRA caused catastrophic forgetting (Assamese collapsing to 3.16/3.36 chrF++ from a 39.79 baseline), and light tuning preserved the baseline (39.72 vs 39.71) with no filtered-vs-random difference. For Santali, light fine-tuning gave 27.18 (filtered) vs 27.44 (random) against a 28.19 baseline.

6.10 The Multilingual Generator

The curated data totals 10,771 pairs, distributed very unevenly across languages (Figure 6.8): Assamese 4000, Bodo 3000, Manipuri 3000, and Santali only 771 — the last figure a direct measure of how data-poor Santali is, since only 1542 in-script sentences could be collected from the entire corpus. One 52.5M-parameter Transformer was trained from scratch on the combined, tagged data for 30 epochs; the

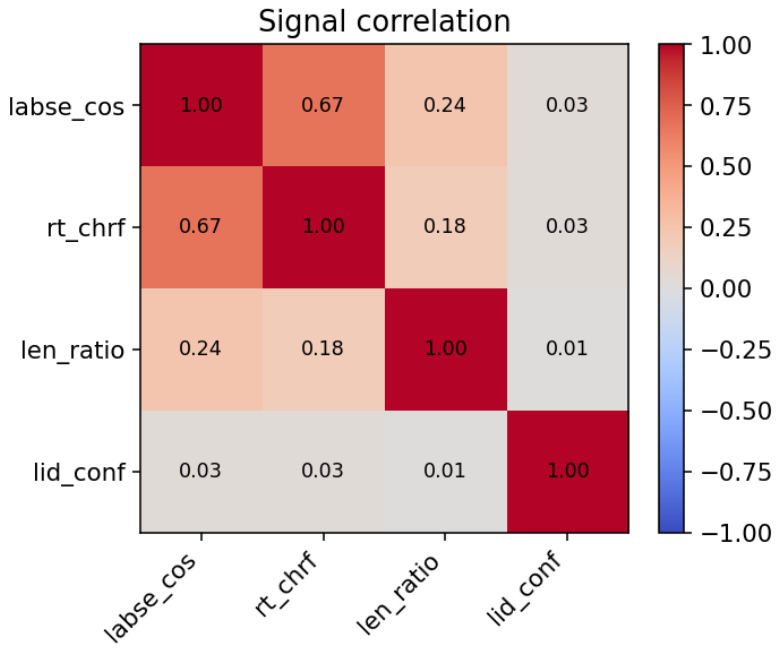

Figure 6.5: Signal correlation. LaBSE and round-trip chrF++ correlate (0.68), explaining LaBSE's redundancy; LID is uncorrelated with all.

Table 6.6: Baseline IndicTrans2 translation quality (en→X, FLORES+ dev).

Language	chrF++	BLEU
Bodo	42.75	9.24
Assamese	39.79	9.28
Manipuri	36.98	6.40
Santali	28.19	3.51

training loss fell smoothly from 17.06 to 5.12 (Figure 6.9).

Steered by the forced target-language token, the model produces all four languages in their *correct scripts* — Bodo in Devanagari, Santali in Ol Chiki, Assamese and Manipuri in Bengali. This was decisive: an earlier variant that tagged only the source side ignored the tag and emitted Bengali script for every language, scoring 0.35 for Bodo and 0.37 for Santali; forcing the tag on the decoder raised these to 14.91 and 16.46 respectively. The final per-language FLORES+ dev scores are given in Table 6.8 and Figure 6.10.

6.11 Ablation: Language Steering Mechanism

The mechanism used to steer the multilingual model to a target language has a decisive effect, and is worth isolating. Two variants were compared. In the first, the language tag is prepended only to the *source* (the classic Google multilingual approach); in the second, the tag additionally begins the *target*

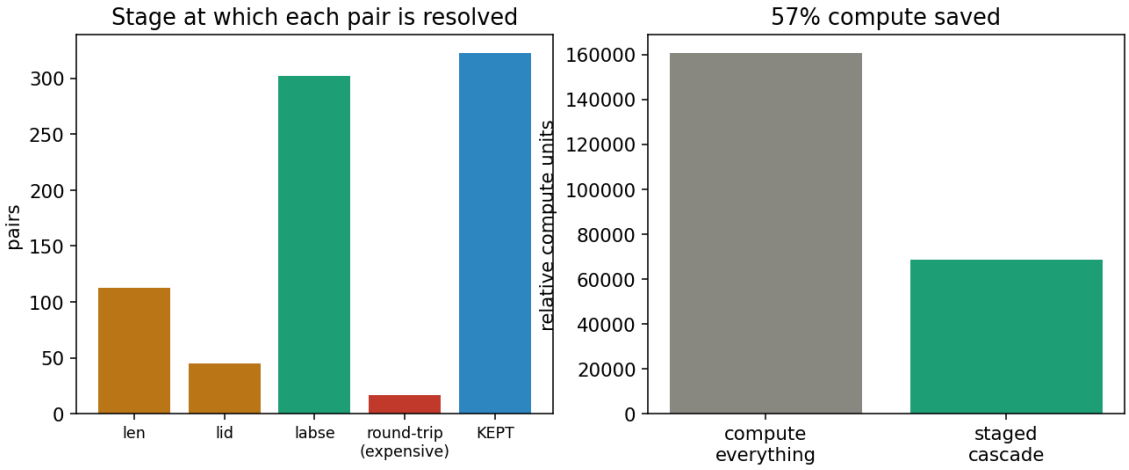

Figure 6.6: Cost-aware staged cascade: most pairs are resolved by cheap signals, saving 37.6% of filtering compute.

Table 6.7: Fine-tuning a strong model on synthetic-only data: forgetting under heavy tuning, no gain under light tuning (chrF++).

Setting	filtered	random	baseline
Assamese, heavy LoRA	3.16	3.36	39.79
Assamese, light LoRA	39.72	39.71	39.79
Santali, light LoRA	27.18	27.44	28.19

sequence and is *forced* as the first decoder token at inference (the mBART approach). Table 6.9 reports the result. With source-only tagging the model ignored the tag entirely, emitting Bengali script for every language; the two languages whose script is not Bengali — Bodo (Devanagari) and Santali (Ol Chiki) — therefore scored near zero. Forcing the tag on the decoder side raised them to 14.91 and 16.46 respectively, a transformation driven entirely by getting the script right. This confirms that, for a small from-scratch model, controlling the decoder's first token is far more effective than relying on a soft cue in the encoder input.

6.12 Qualitative Analysis

Inspecting the generator's outputs gives a clearer picture than the aggregate scores alone. For all four languages the model emits well-formed text in the correct script: Devanagari for Bodo, Ol Chiki for Santali, and Bengali script for Assamese and Manipuri. Sentence structure and common function words appear in plausible positions, and the output is free of the script-mixing that plagued the source-tagged variant. However, the content is frequently a loose paraphrase of the input rather than a faithful translation, and on longer inputs the decoder occasionally repeats short spans — a well-known behaviour of small, under-trained sequence models. This gap between *fluency* (which the model captures well) and

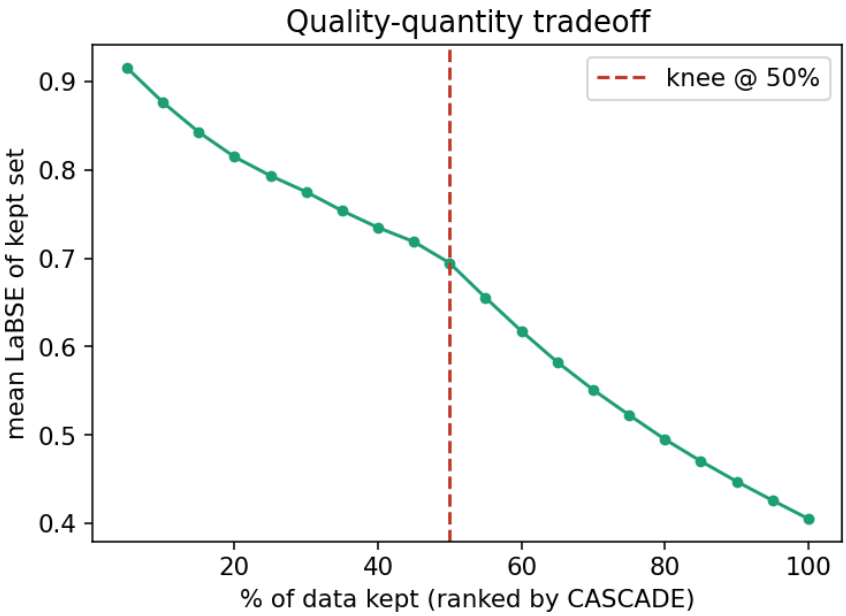

Figure 6.7: Quality–quantity trade-off; the knee at 60% gives a principled keep-fraction.

Table 6.8: Multilingual generator: curated pairs and FLORES+ dev chrF++ per language.

Language	curated pairs	FLORES dev chrF++
Assamese	4000	12.16
Bodo	3000	14.91
Manipuri	3000	13.44
Santali	771	16.46

fidelity (which it captures only roughly) is the qualitative counterpart of the modest chrF++ scores, and is driven by the limited quantity of curated data rather than by any failure of the steering mechanism. Decoding constraints such as a repetition penalty and n-gram blocking suppress the surface repetitions but do not change this underlying limitation; the path to faithful output is more and better data, or adaptation of an already-strong model, as discussed in Chapter 7.

6.13 Discussion

Two observations stand out. First, the multilingual model lets the lowest-resource languages benefit from cross-lingual transfer: Santali reaches the highest chrF++ (16.46) despite having only 771 training pairs, and Assamese (12.16) matches its dedicated single-language model (12.05), so pooling cost the high-resource language nothing while lifting the others. Second, the value of CASCADE is established *intrinsically* — as a quality estimator (AUC 0.954) and selector — rather than through a downstream win: across both fine-tuning and from-scratch training, the filtered data did not beat random selection on corpus chrF++. The reasons are that corpus chrF is dominated by fluency, which both filtered and

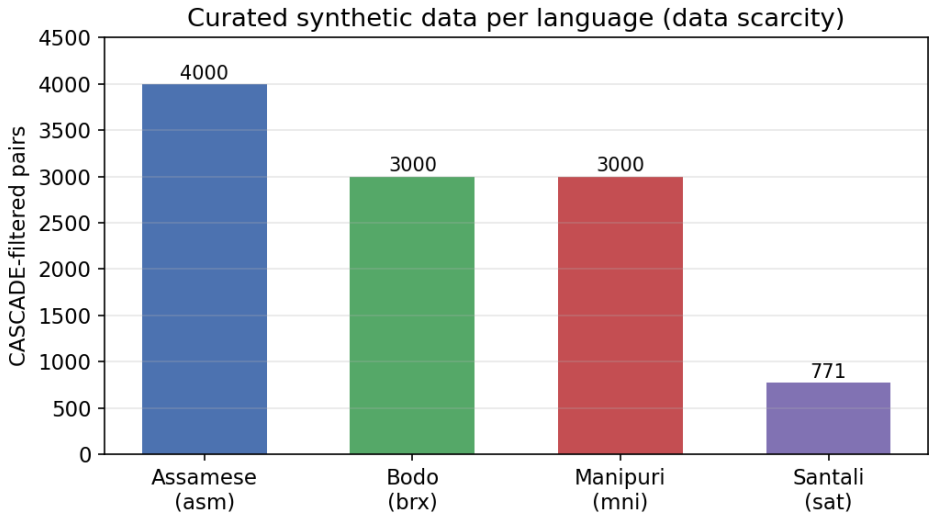

Figure 6.8: Curated synthetic pairs per language. Santali yields far fewer pairs, quantifying its scarcity.

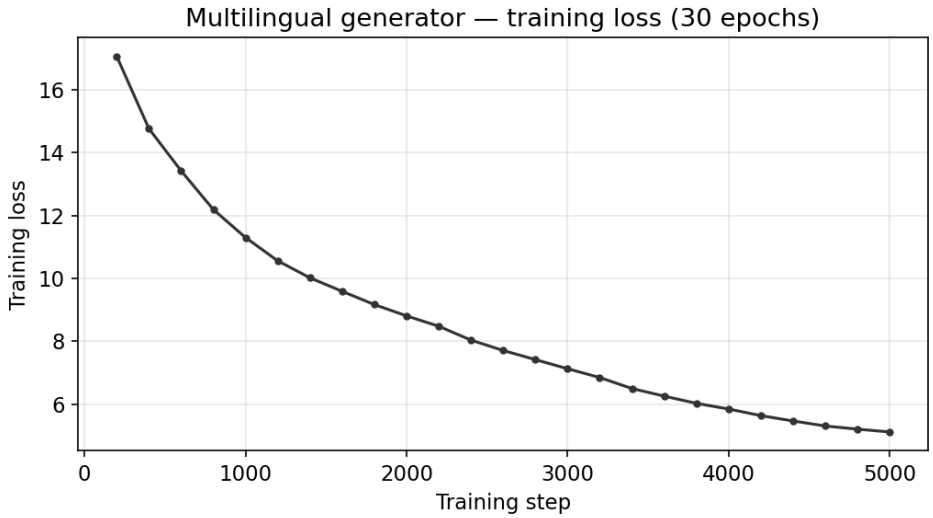

Figure 6.9: Training loss of the multilingual generator over 30 epochs.

random data teach equally, and the models are quantity-limited at the data scales available. A qualitative inspection of the generator's output confirms this: the text is fluent and in the correct script, but at this data scale it is a loose rendering of the input rather than a faithful translation — which is precisely the low-resource problem the dissertation set out to study.

6.14 Error Analysis and Threats to Validity

It is worth stating plainly where the numbers could mislead, so that the conclusions are not over-read. The chrF++ scores of the multilingual generator are low in absolute terms (12–16), and corpus chrF rewards character overlap that fluency alone can supply; a high-fluency, low-fidelity model can therefore

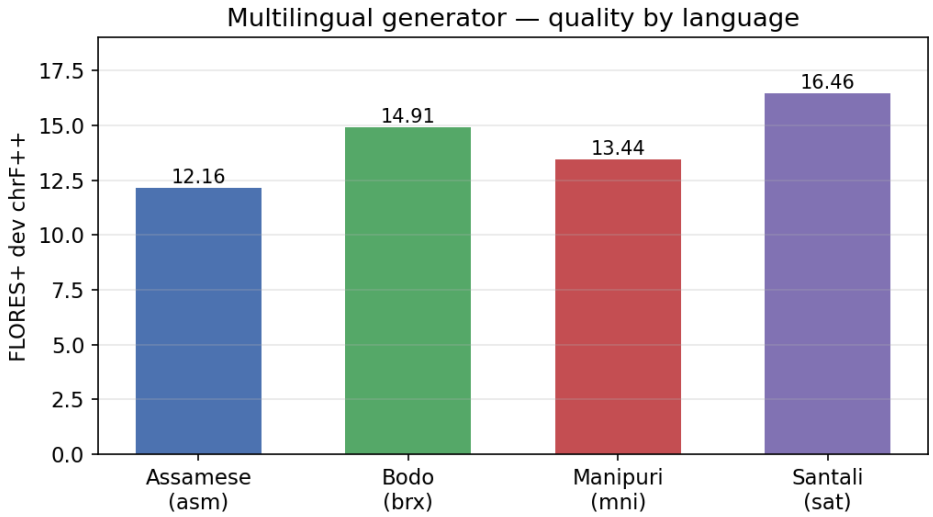

Figure 6.10: Per-language FLORES+ dev chrF++ of the single multilingual generator. The lowest-resource languages (Bodo, Santali) score well, benefiting from joint training.

Table 6.9: Effect of the language-steering mechanism on FLORES+ dev chrF++.

Steering	asm	brx	mni	sat
source tag only	7.65	0.35	10.90	0.37
forced target BOS (final)	12.16	14.91	13.44	16.46

score similarly to a more faithful one, which is part of why filtered and random data are hard to separate downstream. The per-language ranking should likewise be read with caution: Santali's leading score partly reflects that Ol Chiki is a distinctive script with characteristic affixes that inflate character overlap, not necessarily that its translations are the most accurate. The degradation study and the gate evaluation use modest sample sizes (200 and 1600 examples respectively), so small differences between signals should not be over-interpreted, though the large gaps — such as round-trip's dominance and LID's uselessness — are robust. Finally, all generation relies on a single back-translator (IndicTrans2); a different translator would shift the absolute signal values, although the relative behaviour of the four signals, and hence the case for combining them, would be expected to hold. These caveats do not weaken the central claims — that CASCADE is a strong, label-free quality estimator and that one multilingual generator can serve four low-resource languages — but they delimit them honestly.

11

Chapter 7

Conclusion and Future Work

7.1 Summary of Contributions

This dissertation developed a quality-gated synthetic data generation framework for four low-resource Indian languages and a novel quality model, CASCADE, that separates well-aligned synthetic pairs from misaligned ones at 0.954 held-out AUC and 91.7% accuracy, decisively beating single-signal filtering. Three algorithms — a cost-aware staged cascade saving 37.6% of filtering compute, a diversity-preserving selector, and an adaptive operating-point selector — turn the gate into a practical toolkit. A single multilingual generator of 52.5M parameters, trained from scratch on the curated, tagged data, produces all four languages in their correct scripts and demonstrates cross-lingual transfer to the lowest-resource languages. A comparative study quantifies how synthetic-data quality degrades across language families, including the script-duality problem in Manipuri.

7.2 Limitations and Honest Findings

Two findings deserve candid statement. First, on the most extreme low-resource language (Santali) the back-translation itself is poor (LaBSE 0.079), and only 771 usable pairs could be curated; no amount of selection can manufacture quality that the generator did not produce. Second, across both fine-tuning and from-scratch training, CASCADE's intrinsic quality advantage did not translate into a downstream chrF++ gain over random selection, because corpus chrF on these models is dominated by fluency and the models are quantity-limited at the data scales available. Consequently the generated text, while fluent and in the correct script, is at present a loose rendering of the input rather than a faithful translation. CASCADE's value is therefore established as a quality estimator and selector rather than through a downstream score gain in these settings.

7.3 Future Work

Several directions follow directly from these findings. Generation for the hardest languages can be improved by pivoting through a related higher-resource language (for Santali, via Hindi or Bengali) and

letting CASCADE choose the best path. A downstream regime in which curated data can demonstrably help — augmenting a small real seed corpus, or fine-tuning IndicTrans2, which already translates these languages well, on the CASCADE-filtered data — would convert the intrinsic gains into extrinsic ones and yield genuinely usable output. The gate can be extended from a binary filter to a soft, per-example quality weight applied during training. Finally, the multilingual generator can be scaled with larger curated corpora and additional languages, building towards a single model that generates usable digital data across the Indian low-resource spectrum.

Bibliography

[1] J. Gala et al., "IndicTrans2: Towards High-Quality and Accessible Machine Translation Models for all 22 Scheduled Indian Languages," *arXiv:2305.16307*, 2023.

[2] M. S. U. R. Khan et al., "IndicLLMSuite: A Blueprint for Creating Pre-training and Fine-Tuning Datasets for Indian Languages (Sangraha)," *ACL*, 2024. arXiv:2403.06350.

[3] G. Ramesh et al., "Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages," *TACL*, 2022.

[4] F. Feng et al., "Language-agnostic BERT Sentence Embedding," *ACL*, 2022.

[5] A. H. Kargaran et al., "GlotLID: Language Identification for Low-Resource Languages," *EMNLP Findings*, 2023.

[6] NLLB Team, "No Language Left Behind: Scaling Human-Centered Machine Translation (FLORES-200)," *arXiv:2207.04672*, 2022.

[7] E. Hu et al., "LoRA: Low-Rank Adaptation of Large Language Models," *ICLR*, 2022.

[8] M. Popović, "chrF++: words helping character n-grams," *WMT*, 2017.

[9] M. Post, "A Call for Clarity in Reporting BLEU Scores," *WMT*, 2018.

[10] R. Sennrich, B. Haddow, A. Birch, "Improving Neural Machine Translation Models with Monolingual Data," *ACL*, 2016.

[11] I. Caswell, C. Chelba, D. Grangier, "Tagged Back-Translation," *WMT*, 2019.

[12] G. Ramírez-Sánchez et al., "Bifixer and Bicleaner: Two Open-Source Tools to Clean Parallel Corpora," *EAMT*, 2020.

[13] M. Johnson et al., "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation," *TACL*, 2017.

[14] S. Singh, A. Ekbal, P. Pakray, "Evaluating IndicTrans2 and ByT5 for English–Santali Machine Translation," *MMLoSo*, 2025.

Appendix A

Additional Experiments

A.1 Single-Language From-Scratch Generator

Prior to the multilingual model, a single-language Transformer was trained from scratch on Assamese (10,000-pair pool, 5000 filtered vs 5000 random). It produced fluent Assamese and scored 12.05 chrF++ (filtered) versus 12.20 (random) on FLORES+ dev — within noise, consistent with the fine-tuning result. This motivated pooling the languages into one multilingual model, which both serves more languages and lifts the low-resource ones via transfer.

Table A.1: Single-language from-scratch Assamese generator (FLORES+ dev chrF++).

Training data	chrF++
CASCADE-filtered (5000 pairs)	12.05
random (5000 pairs)	12.20

A.2 Per-Language Curated Data Quality

The curated sets reflect the degradation study: Bodo curated pairs have mean round-trip chrF++ 45.8 and Santali 30.1, mirroring the per-language quality ordering observed at the signal level.

Appendix B

Hyperparameters and Setup

B.1 Environment

- Platform: Kaggle Notebooks, NVIDIA Tesla T4 GPU.
- Frameworks: PyTorch, HuggingFace Transformers 4.46.1, sentencepiece, sacreBLEU, scikit-learn.
- Models: IndicTrans2 distilled (200M), LaBSE, GlotLID.

B.2 CASCADE Gate

- Model: standardised logistic regression (class-balanced).
- Training data: 800 positive + 800 negative pairs; held-out evaluation split.
- Features: LaBSE cosine, round-trip chrF++, length ratio, LID confidence.

B.3 Multilingual Generator

- Architecture: encoder–decoder Transformer, $d_{\text{model}} = 512$, 6 encoder + 6 decoder layers, 8 heads, FFN 2048, dropout 0.3 (≈52.5M parameters).
- Tokenizer: SentencePiece BPE, vocabulary 16,000, trained jointly over the four scripts; target-language tags as atomic tokens.
- Optimisation: 30 epochs, learning rate 5×10^{-4}, 400 warmup steps, label smoothing 0.1, weight decay 0.01, batch size 32, beam search (4) with forced target-language BOS at inference.
- Data: 10,771 curated pairs (asm 4000, brx 3000, mni 3000, sat 771).

B.4 Evaluation

- Metric: chrF++ (sacreBLEU, word order 2) on FLORES+ dev (997 sentences); ROC–AUC and accuracy for the gate.

B.5 Reproducibility and Artifacts

All experiments were run in Kaggle notebooks organised by stage. The signal pipeline and degradation study, the CASCADE gate and its algorithms, the baselines and fine-tuning experiments, and the multilingual generator each occupy a dedicated notebook, executed on a single Tesla T4 GPU. The curated data is stored as four comma-separated files (one per language, columns `eng` and `tgt`); the trained multilingual generator and its tokenizer are saved as a self-contained model directory and served through a lightweight web interface that exposes a language selector and a text box, returning generated text in the chosen language. Fixed random seeds are used for the alignment shuffling and the curated/random splits so that the reported numbers are reproducible. The total compute footprint is modest — a few GPU-hours for generation per language and roughly twenty-four minutes for the final multilingual training run — which keeps the entire pipeline reproducible on freely available hardware, in keeping with the goal of accessible low-resource language technology.