Hindi-Chhattisgarhi Cross-Lingual Transfer Using RL-Guided Distillation

Raghav Borikar : M24DS010 : raghavbori@iitbhilai.ac.in
Vikrant Sahu : P24CS007 : vikrantsahu@iitbhilai.ac.in

March 15, 2025

1 Project Overview

This approach focuses on developing a specialized framework that efficiently transfers knowledge from Hindi (high-resource language) to Chhattisgarhi (low-resource language) using reinforcement learning and knowledge distillation techniques. We concentrate resources on optimizing core transfer learning mechanisms.

2 Key Components

- **NLLB-Enhanced Transfer Learning:** Utilizing the No Language Left Behind dataset as the foundation.
- **Reinforcement Learning for Selective Knowledge Transfer:** Using RL to dynamically determine optimal transfer parameters.
- **Bidirectional Lexical Mapping:** Leveraging existing Hindi-Chhattisgarhi parallel data.
- **Knowledge Distillation:** Implementing confidence-guided distillation techniques.

3 Project Timeline

3.1 Phase 1: Setup and Foundational Development

- **Environment Setup and Data Collection**
  - Configure development environment with required libraries and frameworks.
  - Access and prepare the NLLB dataset components relevant to Hindi.
  - Collect and organize existing Hindi-Chhattisgarhi parallel corpus (40,000 sentences).
  - Set up evaluation metrics and benchmarks for cross-lingual transfer.
- **Base Model Architecture Design**
  - Design and implement the base encoder-decoder architecture using NLLB Hindi components.
  - Establish baseline model performance metrics.
  - Create data preprocessing pipelines for both Hindi and Chhattisgarhi corpora.
- **Knowledge Distillation Framework Implementation**
  - Develop teacher and student model architectures for knowledge distillation.
  - Implement confidence-guided distillation mechanisms based on DRL-Rec principles.
  - Design output-level distillation using list-wise KL divergence loss.
  - Implement intermediate-level distillation using Hint loss.

3.2 Phase 2: RL Framework Development

- **RL Environment Design**

  - Define state space (model parameters, performance metrics).
  - Define action space (parameter selection for transfer: freeze/fine-tune/replace).
  - Implement reward function based on translation quality metrics (BLEU, METEOR).

- **Exploring and Filtering Module Implementation**

  - Develop the exploring mechanism to identify valuable transfer candidates.
  - Implement filtering strategies to select informative training instances.
  - Design confidence-guided filtering based on teacher model certainty.

- **Policy Network Development and Integration**

  - Implement RL policy network using Proximal Policy Optimization (PPO).
  - Integrate policy network with the transfer learning framework.
  - Establish monitoring metrics for RL agent performance.

3.3 Phase 3: Training and Optimization

- Initial Model Training and Transfer Strategy Optimization.

- RL-Guided Distillation Process and extensive ablation studies.

3.4 Phase 4: Evaluation and Documentation

- Comprehensive evaluation on Hindi-Chhattisgarhi test sets.

- Drafting research paper and final documentation.

4 Project Repository Structure

```
rich-cousin-poor-sister/
|-- README.md
|-- LICENSE
|-- requirements.txt
|-- .gitignore
|-- data/
|   |-- raw/
|   |-- processed/
|   |-- embeddings/
|   |-- parallel_corpus/
|   |-- evaluation/
|-- src/
|   |-- models/
|   |-- preprocessing/
|   |-- training/
|   |-- distillation/
|   |-- reinforcement_learning/
|   |-- evaluation/
|   |-- utils/
|-- experiments/
|   |-- logs/
```

```
|   |-- checkpoints/
|   |-- configs/
|-- results/
|   |-- figures/
|   |-- tables/
|   |-- metrics/
|-- docs/
|   |-- paper/
|   |-- methodology/
|   |-- tutorials/
|-- tests/
|   |-- unit_tests/
|   |-- integration_tests/
```

5 Expected Outcomes and Novel Contributions

- **First RL-Guided Transfer Learning System for Hindi-Chhattisgarhi.**

- **Enhanced Distillation Techniques.**

- **NLLB Adaptation for Non-Included Languages.**

- **Quantitative Analysis of Hindi-Chhattisgarhi Transfer.**

By focusing resources on core components, this approach has the goal of supporting the "poor sister" language through knowledge transfer from its "rich cousin" while offering a streamlined and focused implementation path.