Team Horizon at BHASHA Task 2: Fine-tuning Multilingual Transformers for Indic Word Grouping

Manav Dhamecha, Gaurav Damor, Sunil Choudhary, Pruthwik Mishra
{u24ai034, u24ai026, u24ai063, pruthwikmishra}@aid.svnit.ac.in

Abstract

Word Grouping involves the identification of a cohesive sequence of words into semantically meaningful units. We present Team Horizon's approach to BHASHA Task 2: Indic Word Grouping. We model the word-grouping problem as a token classification problem and fine-tune multilingual Transformer encoder-only models for the task. We evaluate MuRIL, XLM-Roberta, and IndicBERT v2 and report Exact Match accuracy on the test data.Our best model (MuRIL) achieves **58.1818%** exact match accuracy on the test set and ranks **1st** among all participating teams.

1 Introduction

Word groups often called "Local Word Groups (LWG)" are semantically cohesive units (Karthika et al., 2025) consisting of a sequence of words that convey a single and complete meaning. It is particularly an important characteristic of most Indian languages those belong mainly to the Indo-Aryan and Dravidian families. Word groups can be realized in different forms such as noun compounds, noun groups followed by post-positions, verb groups containing auxiliary verbs, gerund verb groups, light verb constructions using adjectives as the head words.

The concept of local word groups is deeply rooted in the Indian grammatical tradition and well formulated by Panini. This is integrated in the computational paninian framework of sentence level parsing (Akshar et al., 1995). Although Indian languages are free word ordered languages where the constituent or local word groups can move freely in a sentence, the order of words in a group is fixed. Most of the previous works for word grouping can be broadly categorized into either rule-based (Singh et al., 2012) or data-driven. We model this task as a sequence classification problem. To identify word groups, we use the BIO (Tjong Kim Sang, 2002) annotation scheme largely followed in sequence labeling tasks such as constituency parsing, chunking, and named entity recognition. As token classification tasks are successfully modeled using the transformer architecture, we also follow the same strategy for this shared task. Our contributions are as follows:

- A simple and effective BIO token-classification pipeline for Indic word grouping.
- Fine-tuning and evaluation of three multilingual pretrained encoders (MuRIL, XLM-R, IndicBERT v2) with a class-weighted loss to mitigate the dominant O-label bias.
- A concise error analysis highlighting model strengths and typical failure modes, and practical hyperparameters extracted from our implementation.

2 Task Description

BHASHA Task 2 (Indic Word Grouping) (**?**) requires systems to reorder/join tokens into correct word-groupings. The official evaluation metric used in this shared task is **Exact Match Accuracy**: a prediction is correct only if the entire grouped output sentence matches the gold grouped sentence exactly.

3 Methodology

3.1 Problem framing

We treat grouping as token classification with three labels {B, I, O}. Input sentences are tokenized using the pretrained model tokenizer. The tokens after the tokenization steps are actually subwords that are obtained using either wordpiece (Song et al., 2021) or sentencepiece (Kudo and Richardson, 2018). The training data usually consists of words and its corresponding labels. Hence, the labels need to be aligned with subwords after tokenization. This is performed as a pre-processing step in our task.

3.2 Models

We fine-tune three pretrained Transformer encoders using HuggingFace's `AutoModelForTokenClassification`:

- MuRIL (Khanuja et al., 2021) — strong coverage for Indian languages.
- XLM-Roberta (Conneau et al., 2020) — multilingual encoder trained on large multilingual corpora.
- IndicBERT v2 (Doddapaneni et al., 2023) — Indic-specific model (MLM-pretrained).

3.3 Weighted loss to address class imbalance

Word-grouping datasets typically have many tokens aligned to the 'O' label (delimiters), producing an 'all-O' bias. We compute simple inverse-frequency class weights from the training labels and use a custom "weighted" loss wrapper around the standard cross-entropy to slightly upweight B and I labels during training. This is described briefly in our implementation and empirically improved token recall for B/I labels.

3.4 Decoding and Reconstruction

Following token-level prediction, we convert predicted label ids to BIO tags and then reconstruct grouped sentences by concatenating wordpieces labelled as the same group; groups are joined with the separator "`__`" (this mirrors the submission format in our pipeline). Exact-match computation compares the reconstructed grouped sentence with the gold grouped sentence.

4 Implementation and Training Details

We implemented the word group identification task using the Huggingface (Wolf et al., 2020) framework that provides a unified API for different transformer architectures. The hyper-parameters used in training are presented in Table 1. For training the models, we utilize a H100 NVIDIA GPU with 94GB RAM.

5 Dataset and Data Preparation

We followed the official BHASHA/IndicWG dataset layout and used the provided train, dev, and test splits. Table 2 reports detailed statistics for each split.

Parameter	Setting
Label Map	`{B:0, I:1, O:2}`
Optimizer	AdamW
Learning Rate	3×10^{-5}
Epochs	20
Batch Size	8 (train/eval)
Weight Decay	0.01

Table 1: Training configuration and hyperparameters.

The dataset shows moderate variability in length: training inputs average ≈141 characters and ≈30 words, while grouped outputs are slightly longer in characters (because of inserted '__' markers) but have fewer grouped tokens (multiple words merged into single units). Dev and test splits follow similar trends, with dev examples slightly longer on average.

Label construction and token alignment. We converted the grouped outputs (which use the '__' token to indicate a group boundary) to BIO labels. We used the tokenizer's `word_ids()` helper to align word-level labels to subword tokens. During preprocessing, we also apply:

- normalization of punctuation (consistent Unicode forms),
- trimming and collapse of extra whitespace,
- normalization of repeated punctuation or special characters.

For subword-token labeling, we adopt the common practice of marking only the first subword of a word with its BIO tag and setting labels for subsequent subwords to `-100` so the loss function ignores them. (Alternatively, you can propagate the same label to all subwords; we recommend `-100` for cleaner training unless you have a reason to propagate labels.)

Data Augmentation. We augment 5000 sentences from a publicly available Hindi annotated dataset (Mishra et al., 2024) [1] with the original data. We evaluated it under the same scheme. The data augmentation is based on a rule based local word group finder [2] that uses chunk labels and POS tags to form noun and verb groups.

[1] https://github.com/ltrc/shallow_parsing_in_indian_languages

[2] https://github.com/Pruthwik/Rule-Based-LWG

Statistic	Train	Dev	Test
Total Sentences	550	100	226
Input Sentence Statistics			
Avg. Character Length	140.91	159.36	151.64
Min. Character Length	26	34	25
Max. Character Length	901	404	619
Avg. Word Count	29.96	33.80	32.50
Min. Word Count	6	7	7
Max. Word Count	190	90	124

Table 2: Comprehensive statistics for the Hindi Word Segmentation dataset across train, dev, and test splits. Output sentences contain word group boundaries marked with __ separators.

Table 4: Official challenge submission vs. post-challenge result.

Setting	Dev EM (%)	Test EM (%)
Challenge submission (official LB)	35.00	45.13
Post-challenge (MuRIL, refined)	46.58	58.18

6 Experiments and Results

6.1 Evaluation metric

We report **Exact Match Accuracy** (in percent) — the official metric for this task — computed on reconstructed grouped sentences.

6.2 Results

The following table summarizes the validation/test exact-match scores obtained for the three models we fine-tuned.

Model	Dev EM(%)	Test EM(%)
MuRIL	**46.58**	**58.1818**
XLM-R	39	53.3636
IndicBERT v2	35.4	52.7272
MuRIL(5K)	21.3	30.58

Table 3: Exact match scores from our fine-tuned systems (reported as percentages).

6.2.1 Challenge submission vs. post-challenge improvements

This system paper accompanies our participation in the shared task. Our best *official challenge submission* achieved **45.13%** exact match on the test set. After the deadline, minor refinements to training and decoding (e.g., class-weighted loss, boundary reconstruction cleanup) yielded a *post-challenge* result of **58.18%** exact match on the same test set (reported in Table 3); this improved score is not part of the official leaderboard.

6.2.2 Augmented model

We train a model using the 5K augmented data and evaluate it under the same scheme. It achieves an exact-match accuracy of 30.58% on the test set.

MuRIL performed best in our experiments, likely due to targeted pretraining on Indian languages and cased vocabulary which helps preserve morpheme and script cues important for grouping.

6.3 Ablations and observations

We performed a small set of ablations during development:

- **Class weighting:** adding inverse-frequency weights improved B/I recall and increased exact match by a small margin (1–2% absolute) compared to an unweighted baseline.
- **Batch size and epochs:** with our batch size of 8, models required more epochs to converge; we used early-stopping behavior based on validation exact-match to avoid overfitting.
- **Tokenization effects:** models that preserve casing and have better Indic vocabularies (MuRIL) produced fewer tokenization-induced errors.

7 Error Analysis

We quantitatively compare our submissions against gold outputs for both dev and test splits using exact-match on reconstructed grouped sentences.

Dev set (N=100) Exact-match (EM): **35.00%** (35/100). Among 65 mismatches:

- Over-merge (more merges than gold): 33/65 (**50.8%**)
- Over-split (fewer merges than gold): 19/65 (**29.2%**)

- Equal group counts but wrong boundaries: 13/65 (**20.0%**)

Length sensitivity (by input word count):

- ≤20 words: **41.67%** EM
- 21–40 words: **40.82%** EM
- >40 words: **18.52%** EM

Matched vs. mismatched averages: 26.29 vs. 37.85 words (124 vs. 178 chars); avg. absolute boundary deviation = **1.28** “__” markers.

Test set (N=226) Exact-match: **45.13%** (102/226). Among 124 mismatches:

- Over-merge: 68/124 (**54.8%**)
- Over-split: 39/124 (**31.5%**)
- Equal group counts but wrong boundaries: 17/124 (**13.7%**)

Length sensitivity:

- ≤20 words: **63.27%** EM
- 21–40 words: **45.99%** EM
- >40 words: **20.00%** EM

Matched vs. mismatched averages: 26.79 vs. 36.93 words (125 vs. 173 chars); avg. absolute boundary deviation = **1.13** “__” markers.

Qualitative observations

1. **Long compounds and MWEs:** frequent boundary shifts or over-merges.
2. **Ambiguous groupings:** equal group counts yet misaligned boundaries.
3. **Rare/OOV forms and proper nouns:** unstable subword splits hinder reconstruction.
4. **Subword label inconsistencies:** off-by-one boundaries due to wordpiece segmentation.

Annotation Inconsistency. Across gold dev+test, several multiword expressions appear grouped in some sentences but ungrouped in others, introducing unavoidable boundary ambiguity. Table 5 shows frequent examples. Apart from the errors in Table 5, the major inconsistency is seen when an adjective forms a word group with a verb chunk with light verbs. In this case, often the word groups are missed in the annotation. The performance of the models are impacted if there is annotation noise.

Phrase	Tokens (n)	Grouped	Ungrouped
होता है	2	29	7
होती है	2	17	3
करते हैं	2	15	8
होते हैं	2	11	3
हो सकता है	3	11	1
हो सकते हैं	3	9	4
करने की	2	8	5
करने के लिए	3	8	4
तौर पर	2	8	1
करता है	2	7	5
दुर्घटनाग्रस्त हो गया	3	1	1

Table 5: Phrases that are grouped in some gold sentences but ungrouped in others (dev+test).

8 Limitations

Our work focuses solely on a token-classification BIO framework, limiting the diversity of modeling approaches explored. Alternative paradigms such as sequence-to-sequence architectures (e.g., mT5, IndicTrans2), in-context learning, or zero-shot prompting with large language models were not investigated and may offer complementary advantages. While class-weighted loss improved MuRIL performance, we did not benchmark XLM-R or IndicBERT v2 under the same refined setup, leaving comparative analysis incomplete. The gold dataset also contains annotation inconsistencies, particularly in multiword expressions and light-verb constructions, which constrains the achievable exact-match accuracy. Furthermore, the method remains sensitive to tokenizer segmentation due to subword label alignment, and performance degrades on long sentences. The rule-based 5K augmented dataset introduced stylistic mismatch that negatively impacted results, and training on an H100 GPU may limit exact reproducibility on smaller hardware.

9 Conclusion and Future Work

We presented a straightforward BIO token-classification approach for Indic Word Grouping and fine-tuned three multilingual encoders. MuRIL achieved the best exact-match score (58.18%) in our experiments. The approach is simple, reproducible from our notebook, and benefits from class-weighting and careful token-to-word alignment.

Future directions:

- Explore ensembles combining MuRIL and XLM-R outputs (voting or reranking).

- Investigate sequence-to-sequence formulations where the model directly produces grouped outputs (possibly alleviating subword-label alignment issues).

- Try larger models or adapters to improve generalisation on long compounds without extensive compute.

- Augment training data by synthetic perturbations that simulate real-world punctuation/whitespace noise.

Acknowledgements

We thank the BHASHA organizers for the shared task and the dataset. Experiments were run using the implementation in the provided `model.ipynb` (train / eval scripts and decoding utilities).

References

Bharati Akshar, Vineet Chaitanya, and 1 others. 1995. *Natural language processing: a Paninian perspective*. PHI Learning Pvt. Ltd.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.

Sumanth Doddapaneni, Divyanshu Kakwani, Simran Khanuja, Pushpak Bhattacharyya, and Sunayana Sitaram. 2023. Indicbert v2 and the revival of classical indic nlp benchmarks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

N J Karthika, Adyasha Patra, Nagasai Saketh Naidu, Arnab Bhattacharya, Ganesh Ramakrishnan, and Chaitali Dangarikar. 2025. Semantically cohesive word grouping in indian languages. *Preprint*, arXiv:2501.03988.

Simran Khanuja, Arijit Dey, Pushpak Bhattacharyya, Sunayana Sitaram, and Monojit Choudhury. 2021. Muril: Multilingual representations for indian languages. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Pruthwik Mishra, Vandan Mujadia, and Dipti Misra Sharma. 2024. Multi task learning based shallow parsing for indian languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(9):1–18.

Smriti Singh, Om P. Damani, and Vaijayanthi M. Sarma. 2012. Noun group and verb group identification for Hindi. In *Proceedings of COLING 2012*, pages 2491–2506, Mumbai, India. The COLING 2012 Organizing Committee.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. 2021. Fast WordPiece tokenization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2089–2103, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.