Robust Named Entity Recognition in Hinglish Code-Mixed Text using Ensemble of Pre-trained Transformers

# Mayur Shirke

*Dept. of Computer Engineering Pune Institute of Computer Technology*

Pune, India [shirkemayur49@gmail.com](mailto:shirkemayur49@gmail.com)

# Madhushri Wagh

*Dept. of Computer Engineering Pune Institute of Computer Technology*

Pune, India [madhushriwagh15@gmail.com](mailto:madhushriwagh15@gmail.com)

# Amey Shembade

*Dept. of Computer Engineering Pune Institute of Computer Technology*

Pune, India [ameyshembade2511@gmail.com](mailto:ameyshembade2511@gmail.com)

# Rajani Jadhav

*Dept. of Computer Engineering Pune Institute of Computer Technology*

Pune, India [rrjadhav@pict.edu](mailto:rrjadhav@pict.edu)

# Pavan Thorat

*Dept. of Computer Engineering Pune Institute of Computer Technology*

Pune, India [thoratpavan64@gmail.com](mailto:thoratpavan64@gmail.com)

# Raviraj Joshi

*Indian Institute of Technology Madras L3Cube Labs, Pune*

Pune, Maharashtra, India [ravirajoshi@gmail.com](mailto:ravirajoshi@gmail.com)

***Abstract*—With the rise of multilingual societies and the everyday use of informal digital communication, code-mixed text, especially mixes like Hindi-English (Hinglish) written in Roman script has become incredibly common on social media platforms like WhatsApp, Facebook, and even in chatbot interactions. Despite this, Named Entity Recognition (NER) for code-mixed language remains a challenging task due to inconsistent grammar, spelling variations, and lack of standard datasets. Traditional NLP models struggle with such noisy data, and while Large Language Models (LLMs) like GPT offer impressive capabilities, they often come with high computational costs, latency issues, and memory constraints, making them less ideal for real-time or resource-limited environments.**

**This paper focuses on addressing NER in code-mixed texts by exploring ensemble-based approaches that combine the strengths of multiple models. Specifically, we investigate four ensemble techniques—soft voting, hard voting, weighted averaging, and a meta-classifier, using outputs from three code-mixed pre-trained models, which are HingBERT, HingRoBERTa, and Hing-mBERT. Our goal is to show that model ensembling can significantly improve performance on code-mixed NER tasks.**

***Index Terms*—Code-Mixed NER, Hinglish, Transformer Mod- els, Ensemble Methods, HingBERT**

1. Introduction

In recent years, the rise of multilingual digital societies has led to a significant increase in the use of code-mixed languages, particularly Hindi-English (Hinglish) written in Roman script, across platforms like WhatsApp, Facebook, and Twitter [2]. These informal forms of communication are especially prevalent in social media interactions and conversa- tional agents. However, processing such noisy, unstandardized text poses several challenges for Natural Language Processing tasks, particularly Named Entity Recognition [1][4].

NER for code-mixed data is uniquely difficult due to in- consistent grammar, frequent spelling variations, and a lack

of annotated corpora [8]. Traditional NLP pipelines often fail to generalize well on these noisy datasets, while even state- of-the-art multilingual models are not always optimized for such cross-lingual and cross-script settings [17]. Several efforts have been made to address these challenges by leveraging multilingual embeddings [3], domain-specific fine-tuning, and multi-channel architectures [11]. However, the performance gap remains notable, particularly in low-resource environments where computational efficiency is critical.

While Large Language Models (LLMs) such as GPT, PaLM, and Mix-of-Experts architectures [9] have shown promise in handling noisy or complex inputs, their adoption in real- time and resource-constrained environments is often hindered by high computational costs, latency issues, and memory limitations. Therefore, there is a pressing need for lightweight yet robust approaches tailored to the unique structure of code- mixed data. Consequently, there is a growing interest in more lightweight and modular solutions, including ensemble learn- ing techniques, which can effectively combine the strengths of multiple models to improve robustness and generalization [6]. In this paper, we focus on improving Named Entity Recog- nition (NER) performance for Hindi-English (Hinglish) code- mixed text through ensemble-based approaches. We build upon transformer models pre-trained on code-mixed datasets, which are HingBERT, HingRoBERTa, and HingMBERT, and explore four ensemble strategies: soft voting, hard voting, weighted averaging, and meta-classification. These methods aim to leverage the diversity of individual models to mitigate their weaknesses and enhance overall accuracy in processing

highly informal, mixed-language social media content.

Ensemble methods, known to improve generalization and robustness, have shown promise in multilingual and code- mixed contexts [10, 11, 14]. Building on these insights, we

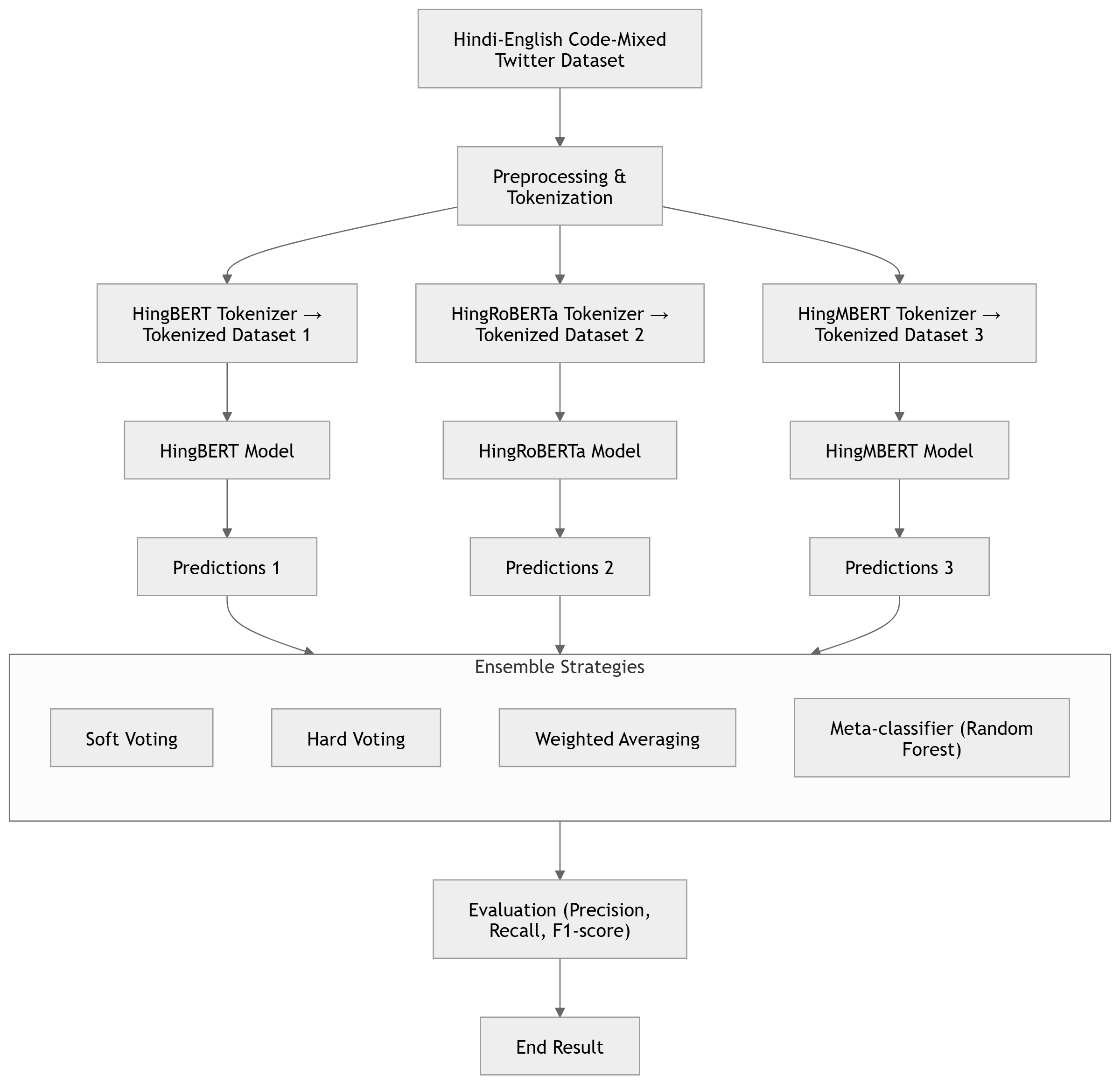


Fig. 1. Flowchart of NER Pipeline with HingBERT, HingRoBERTa, and HingMBERT using Ensemble Methods

evaluate the effectiveness of combining predictions from the three transformer models using the aforementioned ensem- bling techniques.

Our experiments are conducted on a publicly available Hinglish code-mixed NER dataset [8]. We describe the archi- tecture of each model, the ensembling mechanisms, and how the final decisions are derived. Results show that ensemble- based methods consistently outperform individual models. Among the ensemble strategies, weighted averaging yields the best performance on the test dataset, while the meta-classifier performs best on the evaluation dataset. This demonstrates the potential of ensembling mechanisms for building robust and scalable NER systems in low-resource and noisy language settings.

1. Related Work

The task of Named Entity Recognition (NER) in code- mixed languages, especially Hindi-English, has gained sig- nificant interest due to the informal and multilingual nature of user-generated content online. These language mixtures pose considerable challenges for traditional NLP systems, primarily because of their inconsistent grammar, frequent code-switching, and non-standard vocabulary usage.

A large-scale effort focused on Marathi-English code- mixing introduces MeCorpus, pretrained language models (MeBERT, MeRoBERTa), and annotated benchmarks for sen- timent, hate speech, and language ID tasks. These resources significantly outperform multilingual baselines, establishing a strong foundation for code-mixed Marathi NLP research [5]. A massive Hindi-English corpus (HingCorpus) comprising over 52 million sentences was constructed to train a family

of BERT-based models. These models, such as HingBERT and HingRoBERTa, demonstrate superior results across several GLUECoS benchmark tasks, confirming the benefits of using real code-mixed data over synthetic alternatives [12].

To enhance the understanding of code-mixed Hindi-English text, this study proposes word-level and sentence-level lan- guage tagging techniques as input augmentations. These augmentations consistently boost classification performance across sentiment, emotion, and hate speech datasets without altering model architecture [16].

A comparative evaluation of various pretrained models including BERT, RoBERTa, mBERT, and code-mixed models like HingBERT reveals that models trained on actual code- mixed data deliver the best results across multiple Hindi- English NLP tasks. This reinforces the need for tailored pretraining in multilingual and noisy text scenarios [13].

Initial explorations into this area focused on understanding the linguistic characteristics of Hindi-English mixed content. Such studies revealed that standard grammatical patterns are often disrupted in these conversations, making conventional NLP techniques ineffective [2]. Attempts to build robust POS taggers using Conditional Random Fields (CRF) for this domain underscored the necessity of steps like token normalization and precise language identification [9].

Further advancements were seen in multilingual NER through tasks like CALCS, where systems utilizing multilin- gual embeddings outperformed traditional rule-based methods when applied to code-mixed datasets such as Spanish-English and Arabic-English [1]. Research also extended into creating evaluation benchmarks like GLUECoS, which tested models including mBERT and XLM-R across various code-switched language tasks and found them particularly effective in man- aging linguistic variation [11].

Deep learning models, particularly BiLSTM-CRF frame- works, gained popularity in NER applications due to their capacity to learn contextual patterns and leverage character- level features, improving performance on noisy and informal texts [15]. The adoption of transformer-based architectures brought further improvements, as these models could be fine- tuned on mixed-language datasets to perform well even under low-resource conditions [17].

For Indian code-mixed data, new resources were introduced with annotated datasets for languages like Tamil-English and Malayalam-English, promoting further experimentation and model training [4].

A more recent and promising approach has been the integra- tion of multiple learning models via ensembling techniques. Hybrid systems combining statistical models like CRFs with rule-based modules showed significant gains by leveraging both syntactic structures and learned patterns [10].

More complex ensemble architectures incorporated multiple neural components—such as CNNs, BiLSTMs, and trans- former layers—into a unified system that captures complemen- tary linguistic features [8]. Another line of work employed ensemble methods across transformer outputs from models like XLM-R, RoBERTa, and mBERT, using voting or weighted

strategies to combine their predictions and enhance robustness and accuracy [14].

Another relevant contribution is by the CMNEROne team in the SemEval 2022 MultiCoNER shared task, where they tack- led the challenges of code-mixed NER using a multilingual BERT model.[7]

A recent study introduced a Bengali-English code-mixed cross-script dataset and proposed a domain-specific NE tax- onomy for social media content. Their models showed strong performance in extracting named entities, which are critical for applications like question answering. These diverse approaches underscore the importance of ensembling and hybridization in improving NER performance for noisy and code-switched texts.[3]

Several distantly supervised models have shown promising results in Named Entity Recognition (NER), but they often suffer from incomplete and noisy annotations. To address this, BOND-MoE employs a Mixture of Experts (MoE) approach, using multiple models ensembled under an Expectation- Maximization framework to mitigate noisy supervision. This method has demonstrated state-of-the-art performance in NER tasks across real-world datasets.[6]

Multilingual Meta-Embeddings (MME) is a method that learns multilingual representations using monolingual pre- trained embeddings without requiring explicit language iden- tifiers. It employs a self-attention mechanism to fuse infor- mation across embeddings effectively. Evaluated on a code- switched English-Spanish NER dataset, MME achieves state- of-the-art results and generalizes well to unseen language tasks.[18]

1. Methodologies

In this work, we approach the problem of Named Entity Recognition (NER) in Hindi-English code-mixed text by first selecting transformer-based models that are specifically fine- tuned for handling Romanized code-mixed language. We use three models—HingBERT, HingRoBERTa, and HingM- BERT—which have shown strong performance on similar tasks and are trained on datasets that closely resemble the kind of informal, mixed-language data found on social media. We use a publicly available code-mixed tweet dataset an- notated with named entities such as Person, Organization, and

Location.

To improve prediction accuracy, we experiment with ensem- ble learning techniques by combining the outputs of our base models using different strategies. Specifically, we apply soft voting, hard voting, weighted averaging, and a meta-classifier approach to assess which combination delivers the best results for this noisy and low-resource task. Each technique is tested and evaluated to understand how model diversity can be used to enhance overall performance.

1. *Models and Datasets*
   1. *Models:* In this work, we use three different transformer- based models that are specifically fine-tuned for Hindi-English code-mixed text written in Roman script. All three models are

trained on the L3Cube-HingCorpus dataset, which contains a large amount of code-mixed data.

* + - **HingBERT:** HingBERT is based on the standard BERT (Base, uncased) architecture. It is fine-tuned on Hindi- English code-mixed Roman script text. This model is designed to better understand the unique structure and mix of Hinglish language found in social media and conversational data.
    - **HingRoBERTa:** HingRoBERTa is based on XLM- RoBERTa, which is a multilingual variant of RoBERTa. Like HingBERT, it is also fine-tuned on the L3Cube- HingCorpus. This model performs well with multilingual and code-mixed data and is robust in understanding context in Roman-script Hinglish.
    - **HingMBERT:** HingMBERT is based on the original multilingual BERT (mBERT) model and fine-tuned on the same L3Cube-HingCorpus. It is optimized to handle the mixed vocabulary and grammar seen in Hindi-English Roman text.
  1. *Datasets:* For this study, we use a publicly available Hindi-English code-mixed tweet dataset 1 specifically curated for the task of Named Entity Recognition (NER). The dataset consists tweets collected over the past eight years and spans a broad spectrum of topics such as politics, sports, social events, and other culturally relevant themes from the Indian subconti- nent. These tweets are written in Roman script, reflecting the casual, conversational tone typical of social media discourse in multilingual societies. The original dataset creators applied rigorous filtering process to retain only the code-mixed Hindi- English tweets. Tweets that were purely in English, written in Devanagari script, or consisted solely of hashtags, URLs, or non-linguistic noise were removed.

TABLE I

Named Entity Tag Distribution in the Hindi-English

Code-Mixed NER Dataset

| **NER Tag** | **Token Count** |
| --- | --- |
| B-Per | 2362 |
| I-Per | 571 |
| B-Org | 1528 |
| I-Org | 96 |
| B-Loc | 795 |
| I-Loc | 31 |
| **Total NE Tokens** | **5383** |

1. *Entity Tagging*

The dataset has been annotated for three types of named entities: **Person**, **Organization**, and **Location**. Each named entity mention is labeled at the word level using a tagging scheme that includes:

* **B-Per**, **I-Per** for person entities
* **B-Org**, **I-Org** for organization entities
* **B-Loc**, **I-Loc** for location entities

1https://github.com/SilentFlame/Named-Entity-Recognition

* **Other** for all tokens that do not belong to any named entity category

The **B-** tag indicates the beginning of a named entity, while the **I-** tag marks the intermediate of a named entity. This brings the total to seven distinct tags.

**Example Tweet:**

Rahul (B-Per) Sharma (I-Per) ne (Other) New (B-Loc) Delhi (I-Loc) mein (Other) Google (B-Org) join (Other) kiya (Other) hai (Other)

*Translation:* “Rahul Sharma has joined Google in New Delhi.”

1. Experiments
2. *Model Setup*

All models discussed previously were fine-tuned on the Hindi-English code-mixed NER dataset introduced earlier. Each model was trained independently using the same con- figuration to ensure fair evaluation.

1. *Ensemble Methods*

To improve performance beyond individual models, we experiment with four ensemble strategies:

* 1. **Hard Voting:** In hard voting, each model independently predicts a class label for every token in the input sequence. The final label assigned to each token is determined by majority voting across the three models. If two or more models agree on the same class, that class is selected. In scenarios where all three models predict different classes (i.e., a tie), we resolve the conflict using a predefined priority order of entity labels based on validation performance.
  2. **Soft Voting:** Soft voting uses the probability distribu- tions produced by each model. For each token, the class- wise probability scores from all models are averaged. The final class label is then chosen as the one with the highest average probability.
  3. **Weighted Averaging:** A variation of soft voting where model predictions are averaged using weights based on their validation performance. More accurate models contribute more to the final decision.
  4. **Meta-Classifier:** In the meta-classifier approach, we treat the outputs of the base models as input features for a secondary machine learning model. Specifically, for each token, we collect either the predicted class labels or the raw logits from the three base models and use them as input to a Random Forest classifier. The Random Forest learns to identify patterns in these outputs and make a more informed final prediction.

We split the dataset into training, test, and validation sets in an 80:10:10 ratio. Identical data splits were used across all experiments for fair comparisons. All experiments were implemented using PyTorch and HuggingFace’s Transform- ers library. Accuracy, precision, recall, and F1-score were employed as evaluation metrics to conduct a comprehensive comparative analysis.

1. Results

To evaluate the effectiveness of ensemble techniques for Named Entity Recognition (NER) in Hinglish code-mixed data, we compared the performance of four ensemble mod- els—Hard Voting, Soft Voting, Weighted Averaging, and Meta Classifier—against three baseline transformer models: Hing- BERT, HingRoBERTa, and Hing-mBERT. The evaluation was carried out on both validation (Eval) and test datasets using standard metrics: Accuracy, Precision, Recall, and F1-Score. The results are summarized in Tables II and III.

The experimental results demonstrate that ensemble meth- ods consistently outperform individual baseline models in Named Entity Recognition (NER) on code-mixed Hinglish datasets. Combining model predictions enhances robustness and generalization, especially in noisy environments, as seen in the improved performance across both the evaluation and test datasets.

Among the ensemble techniques, the **Meta Classifier** achieves the highest F1-Score (**0.8006**) on the Eval dataset. Using a Random Forest Regressor trained on concatenated base model predictions, it effectively learns inter-model re- lationships, improving its ability to handle edge cases. On the Test dataset, **Weighted Averaging** achieves the best F1- Score (**0.8095**). This method assigns higher weights to stronger models, allowing it to adapt dynamically to unseen data and improve generalization.

In contrast, Hard and Soft Voting provide only modest im- provements over base models. These methods rely on majority and probability averaging, respectively, but lack the adaptabil- ity of more sophisticated ensemble techniques, limiting their ability to capture model-specific strengths.

Among baseline models, **Hing-mBERT** excels in Recall, particularly on the test set, where it achieves **0.8221**. This suggests that Hing-mBERT is effective at identifying more entities, although at the cost of some precision. On the other hand, the Meta Classifier demonstrates a trade-off between precision (**0.8630**) and recall (0.7518), making it more con- servative in its predictions.

Finally, ensemble methods show greater stability across datasets compared to baseline models. While base models like HingBERT and HingRoBERTa show variability in F1-Scores between the Eval and Test datasets, ensemble techniques maintain consistent performance, underscoring their better generalization.

In conclusion, ensemble methods, particularly the Meta Clas- sifier and Weighted Averaging, significantly improve perfor- mance over individual models by combining their strengths, making them highly effective for NER tasks in code-mixed environments.

1. Conclusion

In this study, we addressed the challenge of Named Entity Recognition (NER) in code-mixed Hinglish text—a linguisti- cally complex and noisy domain that is increasingly prevalent in informal communication. We proposed and evaluated a

TABLE II

MODEL PERFORMANCE ON **EVALUATION DATASET**

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| --- | --- | --- | --- | --- |
| Hard Voting | 0.9754 | 0.7632 | 0.8074 | 0.7847 |
| Soft Voting | 0.9751 | 0.7566 | 0.8005 | 0.7780 |
| Meta Classifier | 0.9759 | **0.8230** | 0.7828 | **0.8006** |
| Weighted Averaging | **0.9761** | 0.7665 | 0.8074 | 0.7864 |
| HingBERT | 0.9734 | 0.7289 | 0.8051 | 0.7652 |
| Hing RoBERTa | 0.9738 | 0.7450 | 0.7796 | 0.7619 |
| Hing-mBERT | 0.9744 | 0.7328 | **0.8144** | 0.7714 |

TABLE III

MODEL PERFORMANCE ON **TEST DATASET**

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| --- | --- | --- | --- | --- |
| Hard Voting | **0.9747** | 0.8047 | 0.8134 | 0.8091 |
| Soft Voting | 0.9735 | 0.7931 | 0.7983 | 0.7957 |
| Meta Classifier | 0.9750 | **0.8630** | 0.7518 | 0.7940 |
| Weighted Averaging | 0.9752 | 0.8078 | 0.8113 | **0.8095** |
| HingBERT | 0.9739 | 0.7881 | 0.8069 | 0.7974 |
| Hing RoBERTa | 0.9722 | 0.7848 | 0.7831 | 0.7839 |
| Hing-mBERT | 0.9716 | 0.7580 | **0.8221** | 0.7888 |

series of ensemble learning strategies—Hard Voting, Soft Voting, Weighted Averaging, and Meta Classifier—using the outputs of three transformer-based models fine-tuned for code- mixed NER: HingBERT, HingRoBERTa, and Hing-mBERT.

Our findings on both the evaluation and test datasets demon- strate that ensemble methods consistently surpass individual baseline models across accuracy, precision, recall, and F1- score. Among these, the **Meta Classifier** emerged as the top performer on the evaluation set, achieving the highest F1- score (0.8006). This model, which employs a Random Forest Regressor trained on concatenated prediction vectors from the base models, effectively captures inter-model prediction patterns and decision boundaries, allowing for improved clas- sification—particularly in ambiguous or edge-case scenarios. On the test set, **Weighted Averaging** attained the best F1- score (0.8095), showcasing its strength in leveraging model- specific confidence scores and dynamically assigning higher weight to more reliable predictions. These findings validate the adaptability and robustness of ensemble learning, especially in

unseen test conditions.

Although **Hard Voting** and **Soft Voting** demonstrated mod- est gains over individual models, they lacked the dynamic learning capabilities of the Meta Classifier and Weighted Averaging approaches, limiting their effectiveness. Neverthe- less, their performance still highlighted the benefit of model combination even through simple aggregation techniques.

Among the individual models, **Hing-mBERT** showed the best recall on both datasets, with a test recall of 0.8221, making it particularly effective at identifying a higher number of entities, albeit with some compromise on precision. This indicates that while individual models can be strong in spe- cific aspects, they fall short in overall balance compared to

ensemble strategies.

Overall, this work demonstrates that ensemble tech- niques—especially meta-classification and weighted averag- ing—are highly effective in handling the challenges posed by noisy, code-mixed Hinglish text. These methods not only enhance prediction stability across datasets but also address the limitations of standalone models by integrating their strengths. **Future work** can explore dynamic ensemble selection strategies, multilingual ensembling across other low-resource code-switched languages, and real-time deployment opti- mizations for resource-constrained environments. Additionally, combining ensemble learning with error-specific data augmen- tation or domain-adaptive pretraining may further improve

NER performance in highly informal settings.

Acknowledgments

This work was done under the L3Cube Pune mentorship program. We want to thank our mentors at L3Cube for their continuous support and encouragement.

References

1. G. Aguilar, S. Maharjan, and T. Solorio. “Named En- tity Recognition on Code-Switched Data: Overview of the CALCS 2018 Shared Task”. In: *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*. 2018.
2. K. Bali et al. ““I am borrowing ya mixing?”: An Analysis of English-Hindi Code Mixing in Facebook”. In: *Proceedings of the First Workshop on Computational Approaches to Code Switching*. 2014.
3. Somnath Banerjee et al. “Named Entity Recognition on Code-Mixed Cross-Script Social Media Content”. In: *Computacio´n y Sistemas* 21.4 (2017), pp. 861–869.

ISSN: 2007-9737. DOI: 10.13053/cys-21-4-2850. URL:

https://doi.org/10.13053/cys-21-4-2850.

1. B. R. Chakravarthi and et al. “Named Entity Recog- nition for Code-Mixed Indian Social Media Text: A Dataset and Baseline”. In: *Proceedings of the WILDRE Workshop*. 2020.
2. Tanmay Chavan et al. “My Boli: Code-mixed Marathi- English Corpora, Pretrained Language Models and Evaluation Benchmarks”. In: *Findings of the Asso- ciation for Computational Linguistics: IJCNLP-AACL 2023*. ACL. 2023, pp. 242–249. URL: https:// github. com/l3cube-pune/MarathiNLP.
3. Xinwei Chen et al. “Mix of Experts Language Model for Named Entity Recognition”. In: *Proceedings of the 6th International Conference*. IEEE, 2024.
4. Suman Dowlagar and Radhika Mamidi. “CMNEROne at SemEval-2022 Task 11: Code-Mixed Named En- tity Recognition by Leveraging Multilingual Data”. In: *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Seattle, Washing- ton: Association for Computational Linguistics, 2022,

pp. 1601–1606. URL: https : / / aclanthology. org / 2022 . semeval-1.219.

1. A. Ghosh, A. Das, and T. Chakraborty. “Multi-Input Multi-Channel Deep Neural Network for Code-Mixed Named Entity Recognition”. In: *IEEE Access* 8 (2020),

pp. 189512–189523.

1. A. Jamatia, B. Gamba¨ck, and A. Das. “Part-of-Speech Tagging for Code-Mixed English-Hindi Twitter and Facebook Chat Messages”. In: *Proceedings of the In- ternational Conference on Recent Advances in Natural Language Processing (RANLP)*. 2015.
2. A. Joshi, P. Bhattacharyya, and M. J. Carman. “Investi- gations in computational code-mixing”. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2016.
3. S. Khanuja et al. “GLUECoS: An Evaluation Bench- mark for Code-Switched NLP”. In: *Proceedings of the 58th Annual Meeting of the Association for Computa- tional Linguistics*. 2020.
4. Ravindra Nayak and Raviraj Joshi. “L3Cube- HingCorpus and HingBERT: A Code Mixed Hindi- English Dataset and BERT Language Models”. In: *Proceedings of the WILDRE-6 Workshop @LREC2022*. ELRA. 2022, pp. 7–12. URL: https://github.com/l3cube- pune/code-mixed-nlp.
5. Aryan Patil et al. “Comparative Study of Pre-Trained BERT Models for Code-Mixed Hindi-English Data”. In: *arXiv preprint arXiv:2305.15722* (2023). URL: https :

//arxiv.org/abs/2305.15722.

1. T. Pires et al. “Comparing Pretrained Multilingual Models for Code-Switching”. In: *Transactions of the Association for Computational Linguistics* 9 (2021),

pp. 1099–1115.

1. A. Sinha and et al. “A Multilingual Approach to Named Entity Recognition for Indian Languages”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.
2. Gauri Takawane et al. “Language augmentation ap- proach for code-mixed text classification”. In: *Natural Language Processing Journal* 5 (2023), p. 100042. DOI: 10.1016/j.nlp.2023.100042.
3. G. I. Winata et al. “Learning Fast Adaptation on Cross- Accented Speech Recognition”. In: *Proceedings of In- terspeech 2019*. 2019.
4. Genta Indra Winata, Zhaojiang Lin, and Pascale Fung. “Learning Multilingual Meta-Embeddings for Code- Switching Named Entity Recognition”. In: *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. Ed. by Isabelle Augenstein et al. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 181–186. DOI: 10.18653/v1/ W19-4320. URL: https://aclanthology.org/W19-4320/.