# Fine-tuning Pre-trained Named Entity Recognition Models For Indian Languages

**Sankalp Bahad[1], Pruthwik Mishra[1], Karunesh Arora[2]**
**, Rakesh Chandra Balabantaray[3], Dipti Misra Sharma[1]** and **Parameswari Krishnamurthy[1]**
LTRC, IIIT Hyderabad [1],
CDAC Noida [2], IIIT Bhubaneswar [3]
{sankalp.bahad, pruthwik.mishra}@research.iiit.ac.in
karunesharora@cdac.in, rakesh@iiit-bh.ac.in
{dipti, param.krishna}@iiit.ac.in

## Abstract

Named Entity Recognition (NER) is a useful component in Natural Language Processing (NLP) applications. It is used in various tasks such as Machine Translation, Summarization, Information Retrieval, and Question-Answering systems. The research on NER is centered around English and some other major languages, whereas limited attention has been given to Indian languages. We analyze the challenges and propose techniques that can be tailored for Multilingual Named Entity Recognition for Indian Languages. We present a human annotated named entity corpora of ∼40K sentences for 4 Indian languages from two of the major Indian language families. Additionally,we present a multilingual model fine-tuned on our dataset, which achieves an F1 score of ∼0.80 on our dataset on average. We achieve comparable performance on completely unseen benchmark datasets for Indian languages which affirms the usability of our model.

## 1 Introduction

Named entities are usually real world objects that are denoted by proper names such as “Location", “Person”, “Organization", etc. Named Entity Recognition (NER) is defined as a process of classifying each named entity into a category within a given piece of text. NER is very useful in the understanding of the structure and content of the textual information, and it also plays a pivotal role in various NLP applications.

India has a wide range of languages, where each language has a unique structure, script, grammar, and other linguistic characteristics. Considering India’s linguistic diversity, designing accurate and robust NERs for Indian languages bears even greater significance. We also encounter different challenges while working with NER in an Indian language setup, mainly Hindi, Urdu, Telugu and Odia. These challenges mainly arise due to the following reasons:

1. **Absence of Fixed Word Order**: Indian languages are free word ordered languages, where words can be moved around without changing the meaning of the sentence.

2. **Absence of Capitalization**: Indian language scripts do not have capitalization which makes it difficult to recognize the proper nouns in a sentence or phrase unlike English and other European languages.

3. **Spelling Variations**: Many Indian languages show the property of variations in spellings of the words.

4. **Variation in Word Senses**: In Indian languages, a single word can have multiple meanings based on its sense of usage. This might lead to a case where a word might belong to two different named entities, which can only be determined based on the context.

The emergence of models such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) and many of its variants has added a new dimension to NER with the possibility of developing multilingual NER solutions. This was made possible due to the training data of these models, that consisted of multiple languages. These models, unlike traditional machine learning models, demonstrated the ability of knowledge transfer across languages. This made NER more adaptable and accessible to low resource languages, like many of the Indian languages, which are still largely unexplored and low resourced.

Many Indian languages suffer from lack of labelled data, linguistic resources, and NLP toolkits which is required for designing specific language related features for most of the machine learning models. This issue can easily be resolved by the multilingual neural models by offering a viable solution of knowledge transfer from high to low



*Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pages 75–82

resource languages. Fine-tuning a single multilingual model can leverage the linguistic knowledge encoded with the model. We experiment with different multilingual pre-trained models and show their efficacies with a strong focus on the availability of resources.

## 2 Related Work

The previous works in this field of NER have mainly explored the challenges and opportunities of NER techniques in multilingual settings. Researchers have developed and fine tuned some multilingual NER models, that help perform NER across multiple languages (Nothman et al., 2013). These models rely on pre-trained transformer based architectures, for example: BERT, RoBERTa (Zhuang et al., 2021), XLM-RoBERTa (Conneau et al., 2020). It has been observed that cross lingual transfer learning is extremely useful and effective for low resource languages, where NER models pre-trained on high resource languages are adapted for low resource languages. The research has also focused on creating and curating multilingual corpora encompassing a large range of languages, that prove to be valuable resources for training and evaluating multilingual NER models.

There has been significant amount of work regarding datasets and other resources using pre-trained transformer models. Naamapadam (Mhaske et al., 2023) and HiNER (Murthy et al., 2022) are two widely used publicly available datasets for Indian language NER.

1. Naamapadam Dataset: Naamapadam consists of data from 11 major Indian languages from two language families. The dataset contains more than 400k sentences annotated with a total of at least 100k entities from three standard entity categories (Person, Location, and, Organization) for 9 out of the 11 languages. It is a significant resource for NER in Indian Languages.

2. HiNER Dataset: This is another NER dataset by annotating data from the ILCI tourism domain (Jha, 2010) and a subset of the news domain corpus (Goldhahn et al., 2012) in Hindi. This dataset includes a total of 108,608 sentences and 11 tags.

## 3 Named Entity Annotation

For the task of NER, we annotated data from two domains, general and governance. At least 2 annotators with post graduation education were involved in the task for each language. Named entities are annotated for following 4 languages where 3 are from the Indo Aryan family and 1 from Dravidian family (shown in sequence): **Hindi**, **Odia**, **Urdu**, and **Telugu**. For Hindi, 7 annotators were included. The average inter-annotator agreement for all four languages was 0.95, which shows good agreement among the annotators. The agreement scores are evaluated on 200 sentences for each language. We compute Cohen's Kappa measure for this. For Hindi, we compute the average of Cohen's scores among all possible combinations of the raters. Language-wise inter-annotator agreement scores are reported in Table 1. 6 tags were chosen for named entity tagging, which are detailed in Table 2 followed by the examples of Person, Location, and Organization entities in all languages.

| Language | Agreement Score |
|---|---|
| Hindi | 0.96 |
| Odia | 0.94 |
| Telugu | 0.95 |
| Urdu | 0.96 |

Table 1: Language Wise Inter Annotator Agreement Scores

| Tag | Desc | Example |
|---|---|---|
| NEP | Person names | Virat Kohli |
| NEL | Locations | New Delhi |
| NEO | Organization Names | IIT-Delhi |
| NEAR | Artefacts | Taj Mahal |
| NEN | Number | fifteen thousand |
| NETI | Time and Date | 5th December |

Table 2: Named Entity Tags

| Named Entity | Hindi | Telugu | Urdu | Odia |
|---|---|---|---|---|
| Person | राहुल गांधी (Rahul Gandhi) | కేసీఆర్ (KCR) | عمران خان (Imran Khan) | ନବୀନ ପଟ୍ଟନାୟକ (Naveen Patnaik) |
| Location | दिल्ली (Dilli) | హైదరాబాద్ (Hyderabad) | لاہور (Lahore) | ଭୁବନେଶ୍ୱର (Bhubaneswar) |
| Organization | भारतीय रिज़र्व बैंक (Bharatiya Reserve Bank) | తెలంగాణ రాష్ట్ర సమితి (Telangana Rashtra Samiti) | پاکستان کرکٹ بورڈ (Pakistan Cricket Board) | ଓଡିଶା ମିନେରାଲ୍ସ ଇଣ୍ଟରେସନ୍ (Odisha Minerals Corporation) |

Figure 1: Enter Caption

## 4 Methodology

We first explored various datasets and models available for Hindi Named Entity Recognition. As our named entity annotated corpus is annotated with a different tagset, we could not make use of the existing models directly. In this pursuit, we explored different fine-tuning techniques to develop a model tailor-made for our tagset.

We experiment with two approaches for the creation of monolingual models. First approach is to fine-tune a baseline BERT model for our task, and the second approach fine-tunes a BERT based NER model for our task, on our annotated dataset. As our basic model, we select XLM-RoBERTa-Base (Conneau et al., 2020) model, which is a transformer based architecture designed for multilingual natural language understanding tasks. This model is pretrained on a vast multilingual corpus and hence is capable of efficiently handling multiple languages, which makes it well suited for the multilingual NER task. The selection of this model for multilingual NER in Indian languages can be further justified by its strong performance in various NLP tasks and its ability to generalize well across languages. Its multilingual pre-training enables it to capture linguistic nuances in different languages, including those present in Indian languages.

As our main focus had been creating a multilingual model for low resource languages, we found multiple ways of improving the results for NER for low resource languages, some of them are as follows:

- One method involves extending the vocabulary, encoders, and decoders to accommodate target languages and continuing pretraining on the target language. Subsequently, pretraining continues using monolingual data in the target language.

- Another approach is to use alignment models like MUSE or VecMap with bilingual dictionaries to initialize the embeddings of new vocabulary, instead of randomly initializing them.

- An alternative strategy involves cross-lingual and progressive transfer learning, where language model training for low-resource languages begins with a large language model for a high-resource language, including overlapping vocabulary.

- Building extensive corpora from existing parallel data can also be beneficial. This approach enables the creation of high-quality training data for multilingual models and facilitates the training of models for low-resource languages that may lack sufficient training data.

Out of all these available methods, we find the approach that uses cross lingual and progressive transfer learning, to train language models for low resource languages with language model for high resource languages by appending the vocabulary. This method worked well for languages belonging to the same language family.

We also try taking a different approach of converting the scripts from native to roman script and carrying out the experiments on the multilingual model, but it was observed that the model trained on native scripts was performing better than the model trained on the roman scripts. A reason for this behaviour can be the absence of roman scripts for the corresponding native scripts of the language in the training data of the pretrained XLM RoBERTa (Conneau et al., 2020) base model. Hence, no further exploration was done in this direction.

We also evaluated the dataset on the CRF (Lafferty et al., 2001; Patil et al., 2020) model, which as expected did not give a good result due to the fact that it was not a pre-trained model. The major limitation of a CRF model lies in it inability to transfer knowledge for reusability. Hence, we did not continue any exploration in that direction.

## 5 Experiments

Table 3 shows a list of languages and the corresponding number of sentences in their training, testing, and validation datasets. We have released label-wise count for all languages in the Appendix section. As a part of this work, we release annotated datasets of 4 languages with different degrees of morphological richness: Hindi, Urdu, Odia, and Telugu.

| Language | Train | Test | Dev |
|---|---|---|---|
| Hindi | 11076 | 1389 | 1389 |
| Urdu | 8720 | 1096 | 1094 |
| Odia | 12109 | 1519 | 1517 |
| Telugu | 2993 | 384 | 384 |

Table 3: Language Dataset Split in terms of Sentences

| Label | Dev Dataset | | Test Dataset | |
|---|---|---|---|---|
| | Indic NER F1-Score | HiNER F1-Score | Indic NER F1-Score | HiNER F1-Score |
| NEL | 0.68 | 0.68 | 0.73 | 0.84 |
| NEO | 0.38 | 0.40 | 0.31 | 0.42 |
| NEP | 0.77 | 0.68 | 0.69 | 0.64 |
| Micro Avg | 0.60 | 0.59 | 0.55 | 0.66 |
| Macro Avg | 0.61 | 0.59 | 0.57 | 0.63 |
| Weighted Avg | 0.64 | 0.62 | 0.61 | 0.68 |

Table 4: Comparison of F1-Scores for Indic NER and HiNER models on Dev and Test Datasets

Our experiments include reviewing of the earlier methods including Conditional Random Fields and neural based named entity taggers. In this, we analyze the pre-trained models and datasets released as Indic NER model and Naamapadam dataset (Mhaske et al., 2023) and HiNER (Murthy et al., 2022) .

Our experiments include testing Indic NER and HiNER on our annotated dataset, where we record an F1 score between 0.55 to 0.65 for the dev and test sentences of the gold dataset. We refer to our dataset as *gold* dataset and this convention is used in the future tables and figures. These experiments are conducted to visualize the performance of different models and adapting them towards developing a customized model for our gold dataset. As an initial experiment, we test the publicly available models on each other to assess their performance which are reported in Table 4.

We then proceed towards creating a monolingual model for Hindi. Our hypothesis is that a model that is already trained on NER task is expected to outperform the base model with no knowledge about the NER task. We validate our hypothesis by fine-tuning a baseline BERT model (not trained for an NER task) on our annotated dataset and fine-tuning a BERT based NER model (HiNER) on our annotated dataset. This experiment is carried out on all the tags of our dataset. We report accuracies between BERT (Devlin et al., 2019) based NER model and baseline BERT based model. As expected, the model which is a result of fine-tuning on HiNER model performs better than fine-tuning on baseline BERT model.

We then combine all the data from different languages and train a multilingual model. We experiment with changing of scripts i.e converting all the data to the same script before finetuning, to check whether the new model performs better or worse than the original model. We convert all our data to Roman script for this purpose. We then fine-tune the RoBERTa base model on Naamapadam dataset and gold dataset as the part of the comparative study between native script and roman script.

In the fine-tuning approach used, we combine all the training data for all languages and fine-tune the monolingual model on this combined data. We then analyze the performance of each language on the multilingual model.

## 6 Results and Discussion

### 6.1 Review of earlier methods

In this section, we look at the results of the experiments we performed on the existing models. We used the metrics from the Seqeval (Nakayama, 2018) library to calculate F1 Scores and Classification reports.

Table 4 shows the performance of the IndicNER (Mhaske et al., 2023) and HiNER (Murthy et al., 2022) models on the test and dev sets of our datasets. From the scores, we clearly observe that the model is unable to predict the NEO tags appropriately.

Results of the test set of the data released by HiNER on IndicNER model and test set of the data released by AI4Bharat on HiNER model are shown in Tables 5 and 6 respectively.

These results show the quality of our annotated datasets and how the already available NER models perform on this dataset. Our dataset gives decent scores in zero shot tests on the IndicNER and HiNER models. Further experiments include fine-tuning these models on our dataset and analyzing their results.

### 6.2 Building new models

The test results of the baseline BERT model fine-tuned on our annotated Hindi data is shown in the Table 7 and that of the HiNER model fine-tuned

| Label | Precision | Recall | F1-Score |
|---|---|---|---|
| LOC | 0.88 | 0.65 | 0.75 |
| ORG | 0.62 | 0.59 | 0.60 |
| PER | 0.72 | 0.83 | 0.78 |
| **Micro Avg** | 0.82 | 0.67 | 0.74 |
| **Macro Avg** | 0.74 | 0.69 | 0.71 |
| **Weighted Avg** | 0.83 | 0.67 | 0.74 |

Table 5: Indic NER model on HiNER Dataset

| Label | Precision | Recall | F1-Score |
|---|---|---|---|
| LOC | 0.83 | 0.78 | 0.80 |
| ORG | 0.72 | 0.65 | 0.69 |
| PER | 0.86 | 0.80 | 0.83 |
| **Micro Avg** | 0.81 | 0.75 | 0.78 |
| **Macro Avg** | 0.80 | 0.74 | 0.77 |
| **Weighted Avg** | 0.81 | 0.75 | 0.78 |

Table 6: HiNER model on Naamapadam dataset

on our annotated Hindi data is shown in the Table 8. We observe close to an overall F1 score of 0.82 on the baseline BERT model for our dataset, and an overall F1 score of 0.83 on HiNER Model fine-tuned. This supports our assumption of getting a better score on model fine-tuned on an existing NER model than by fine-tuning a bare BERT model.

| Label | Precision | Recall | F1-Score |
|---|---|---|---|
| NEAR | 0.32 | 0.44 | 0.37 |
| NEL | 0.83 | 0.87 | 0.85 |
| NEN | 0.87 | 0.90 | 0.89 |
| NEO | 0.58 | 0.55 | 0.56 |
| NEP | 0.85 | 0.85 | 0.85 |
| NETI | 0.73 | 0.75 | 0.74 |

Table 7: Performance of the baseline BERT model on our dataset

| Label | Precision | Recall | F1-Score |
|---|---|---|---|
| NEAR | 0.19 | 0.28 | 0.22 |
| NEL | 0.88 | 0.92 | 0.90 |
| NEN | 0.85 | 0.89 | 0.87 |
| NEO | 0.60 | 0.57 | 0.59 |
| NEP | 0.81 | 0.85 | 0.83 |
| NETI | 0.75 | 0.80 | 0.78 |

Table 8: Performance of the HiNER model on our dataset

Table 9 shows the comparison between the F1 scores on the Test set, of the baseline BERT model and the HiNER model fine-tuned on our Hindi annotated data.

| Model | F1 Score |
|---|---|
| baseline BERT Model | 0.8205 |
| HiNER Model | 0.8316 |

Table 9: Comparison of F1 Scores between baseline BERT and HiNER Models

The above results show that using an already trained NER model for fine-tuning is better than using a baseline BERT model for fine-tuning in the monolingual Hindi case.

| Test-Dataset | Monolingual | Multilingual (Combined) |
|---|---|---|
| Gold-Hindi | 0.8205 | 0.8105 |
| Gold-Odia | 0.7546 | 0.7715 |
| Gold-Telugu | 0.7632 | 0.7555 |
| Gold-Urdu | 0.8285 | 0.8331 |

Table 10: F1 Scores for a Multilingual Model

Table 10 shows the F1 Scores of different languages on the monolingual and multilingual models for all the four languages on the Gold dataset. We observe the monolingual and multilingual scores to be in the range of 0.75 to 0.83. The multilingual models exhibit an increase in scores for Odia and Urdu, whereas there is a slight dip in the scores for Telugu and Hindi. A possible reason for this can be that Telugu and Hindi belong to different language families. Overall, multilingual models demonstrates comparable results to monolingual models, exhibiting the capability and effectiveness in multiple languages being handled simultaneously.

We also tested our models on Naamapadam test set. The results are not very useful as that IndicNER can only predict 3 tags, whereas our developed model predicts all the 7 tags.

## Acknowledgement

This annotated corpora has been developed under the Bhashini project funded by Ministry of Electronics and Information Technology (MeitY), Government of India. We thank MeitY for funding this work. We sincerely thank the annotators who developed this corpora whose names are added in the appendix.

## 7 Conclusion and Future Work

We introduce a specialized NER dataset tailored for four Indian languages. Our experiments with established NER models on this dataset provide valuable insights for fine-tuning. Our proposed fine-tuning technique paves a way for NER in low resource languages. Techniques such as transfer learning and architectural modifications can further be explored to improve the model. We propose augmenting our dataset with additional annotated sentences. Adding data from other Indian languages can potentially lead to substantial performance improvements.

## References


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 the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Girish Nath Jha. 2010. The TDIL program and the Indian langauge corpora intitiative (ILCI). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Arnav Mhaske, Harshit Kedia, Sumanth Doddapaneni, Mitesh M. Khapra, Pratyush Kumar, Rudra Murthy, and Anoop Kunchukuttan. 2023. Naamapadam: A large-scale named entity annotated data for Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10441–10456, Toronto, Canada. Association for Computational Linguistics.

Rudra Murthy, Pallab Bhattacharjee, Rahul Sharnagat, Jyotsana Khatri, Diptesh Kanojia, and Pushpak Bhattacharyya. 2022. Hiner: A large hindi named entity recognition dataset.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran. 2013. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175.

Nita Patil, Ajay Patil, and BV Pawar. 2020. Named entity recognition using conditional random fields. *Procedia Computer Science*, 167:1181–1188.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

## Appendix

### Data Statistics

Figures 11, 12, 13, and 14 show a list of label counts for Test, Validation, and Train datasets for Odia, Telugu, Hindi, and Urdu language. Tables 15, 16, 17, 18 show a comparative study of the classification reports for Hindi, Telugu, Urdu, and Odia language for the monolingual and multilingual models.

| Label | Test Count | Validation Count | Train Count |
|---|---|---|---|
| NEAR | 24 | 24 | 183 |
| NEP | 59 | 59 | 471 |
| NETI | 64 | 64 | 509 |
| NEL | 87 | 87 | 695 |
| NEO | 35 | 35 | 280 |
| NEN | 8 | 8 | 60 |

Table 11: Odia Data Label Split

| Label | Test Count | Validation Count | Train Count |
|---|---|---|---|
| NEN | 76 | 76 | 606 |
| NETI | 17 | 17 | 130 |
| NEP | 14 | 14 | 110 |
| NEL | 5 | 5 | 13 |
| NEO | 8 | 8 | 57 |
| NEAR | 5 | 5 | 13 |

Table 12: Telugu Data Label Split

| Label | Test Count | Validation Count | Train Count |
|---|---|---|---|
| NEP | 97 | 97 | 774 |
| NETI | 154 | 154 | 1226 |
| NEN | 295 | 295 | 2357 |
| NEL | 93 | 93 | 742 |
| NEO | 60 | 60 | 476 |
| NEAR | 15 | 15 | 112 |

Table 13: Hindi Data Label Split

| Label | Test Count | Validation Count | Train Count |
|---|---|---|---|
| NEL | 106 | 106 | 847 |
| NEN | 213 | 213 | 1700 |
| NETI | 5 | 5 | 31 |
| NEO | 16 | 16 | 126 |
| NEP | 39 | 39 | 303 |
| NEAR | 5 | 5 | 36 |

Table 14: Urdu Data Label Split

### Label Wise Results

| Category | Monolingual | | | Multilingual | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1-score | Precision | Recall | F1-score |
| NEAR | 0.52 | 0.58 | 0.55 | 0.52 | 0.54 | 0.53 |
| NEL | 0.85 | 0.87 | 0.86 | 0.85 | 0.86 | 0.85 |
| NEN | 0.94 | 0.90 | 0.92 | 0.95 | 0.91 | 0.93 |
| NEO | 0.66 | 0.66 | 0.66 | 0.63 | 0.65 | 0.64 |
| NEP | 0.85 | 0.84 | 0.84 | 0.82 | 0.81 | 0.82 |
| NETI | 0.69 | 0.71 | 0.70 | 0.64 | 0.68 | 0.66 |

Table 15: Comparison of Hindi Named Entity Recognition Performance in Monolingual and Multilingual Settings

| Category | Monolingual | | | Multilingual | | |
|---|---|---|---|---|---|---|
| | **Precision** | **Recall** | **F1-score** | **Precision** | **Recall** | **F1-score** |
| NEAR | 0.67 | 0.50 | 0.57 | 0.75 | 0.50 | 0.60 |
| NEL | 0.70 | 0.58 | 0.64 | 0.80 | 0.57 | 0.67 |
| NEN | 0.87 | 0.90 | 0.88 | 0.84 | 0.91 | 0.87 |
| NEO | 0.42 | 0.56 | 0.48 | 0.50 | 0.56 | 0.53 |
| NEP | 0.59 | 0.57 | 0.58 | 0.58 | 0.70 | 0.64 |
| NETI | 0.49 | 0.74 | 0.59 | 0.43 | 0.52 | 0.47 |

Table 16: Comparison of Telugu Named Entity Recognition Performance in Monolingual and Multilingual Settings

| Category | Monolingual | | | Multilingual | | |
|---|---|---|---|---|---|---|
| | **Precision** | **Recall** | **F1-score** | **Precision** | **Recall** | **F1-score** |
| NEAR | 0.33 | 0.20 | 0.25 | 0.50 | 0.40 | 0.44 |
| NEL | 0.82 | 0.80 | 0.81 | 0.78 | 0.76 | 0.77 |
| NEN | 0.96 | 0.90 | 0.93 | 0.98 | 0.90 | 0.94 |
| NEO | 0.39 | 0.37 | 0.38 | 0.49 | 0.47 | 0.48 |
| NEP | 0.77 | 0.64 | 0.70 | 0.84 | 0.62 | 0.71 |
| NETI | 0.58 | 0.78 | 0.67 | 0.67 | 0.89 | 0.76 |

Table 17: Comparison of Urdu Named Entity Recognition Performance in Monolingual and Multilingual Settings

| Category | Monolingual | | | Multilingual | | |
|---|---|---|---|---|---|---|
| | **Precision** | **Recall** | **F1-score** | **Precision** | **Recall** | **F1-score** |
| NEAR | 0.73 | 0.58 | 0.64 | 0.86 | 0.58 | 0.69 |
| NEL | 0.89 | 0.82 | 0.85 | 0.90 | 0.84 | 0.87 |
| NEN | 0.46 | 0.29 | 0.35 | 0.44 | 0.38 | 0.41 |
| NEO | 0.65 | 0.76 | 0.70 | 0.64 | 0.70 | 0.67 |
| NEP | 0.85 | 0.83 | 0.84 | 0.88 | 0.85 | 0.86 |
| NETI | 0.59 | 0.70 | 0.64 | 0.66 | 0.71 | 0.68 |

Table 18: Comparison of Odia Named Entity Recognition Performance in Monolingual and Multilingual Settings

**Details of Annotators**

| Language | Language Expert | Designation | Affiliation |
|---|---|---|---|
| Hindi | Alpana Agarwal | Senior Language Editor | IIIT-Hyderabad |
| | Preeti Pradhan | Senior Language Editor | IIIT-Hyderabad |
| | Nandini Upasani | Senior Language Editor | IIIT-Hyderabad |
| | Naresh Bansal | Senior Language Editor | IIIT-Hyderabad |
| | Vaibhavi Kailash Kothadi | Senior Language Editor | IIIT-Hyderabad |
| | Pranjali Kanade | Language Editor | IIIT-Hyderabad |
| | Kaberi Sau | Senior Language Editor | IIIT-Hyderabad |
| Odia | Prakash Kumar Bhuyan | Linguist | CDAC-Noida |
| | Bigyan Ranjan Das | Project Assistant | IIIT-Bhubaneswar |
| Telugu | Koustubha NS | Senior Language Editor | IIIT-Hyderabad |
| | Sarala Sree Ramancharla | Senior Language Editor | IIIT-Hyderabad |
| Urdu | Mohammed Younus | Language Editor | IIIT-Hyderabad |
| | Mohd. Noman Ali | Language Editor | IIIT-Hyderabad |