HiNER: A Large Hindi Named Entity Recognition Dataset

Rudra Murthy[2], Pallab Bhattacharjee[1], Rahul Sharnagat[3], Jyotsana Khatri[1], Diptesh Kanojia[4], Pushpak Bhattacharyya[1]
[1]CFILT Lab, IIT Bombay, India.
[2]IBM IRL, Bangalore, India. [3]Walmart Labs, USA.
[4]Surrey Institute for People-centred AI, University of Surrey, United Kingdom.
[1]bhattacharjee.pallab9@gmail.com, {jyotsanak, pb}@iitb.ac.in
[2]rmurthyv@in.ibm.com, [3]rdsharnagat@gmail.com, [4]d.kanojia@surrey.ac.uk

Abstract

Named Entity Recognition (NER) is a foundational NLP task that aims to provide class labels like *Person*, *Location*, *Organisation*, *Time*, and *Number* to words in free text. Named Entities can also be multi-word expressions where the additional I-O-B annotation information helps label them during the NER annotation process. While English and European languages have considerable annotated data for the NER task, Indian languages lack on that front- both in terms of quantity and following annotation standards. This paper releases a significantly sized standard-abiding Hindi NER dataset containing 109,146 sentences and 2,220,856 tokens, annotated with 11 tags. We discuss the dataset statistics in all their essential detail and provide an in-depth analysis of the NER tag-set used with our data. The statistics of tag-set in our dataset show a healthy per-tag distribution, especially for prominent classes like *Person*, *Location* and *Organisation*. Since the proof of resource-effectiveness is in building models with the resource and testing the model on benchmark data and against the leader-board entries in shared tasks, we do the same with the aforesaid data. We use different language models to perform the sequence labelling task for NER and show the efficacy of our data by performing a comparative evaluation with models trained on another dataset available for the Hindi NER task. Our dataset helps achieve a weighted F1 score of 88.78 with all the tags and 92.22 when we collapse the tag-set, as discussed in the paper. To the best of our knowledge, no available dataset meets the standards of volume (amount) and variability (diversity), as far as Hindi NER is concerned. We fill this gap through this work, which we hope will significantly help NLP for Hindi. We release this dataset with our code and models for further research.

Keywords: named entity recognition, dataset, Hindi, human-annotated, low-resource language

1. Introduction

Named Entity Recognition (NER) is an essential lower-level task (Ma and Hovy, 2016) in Natural Language Processing (NLP), used to extract and categorize naming entities into a predefined set of classes such as *person*, *location*, *organization*, *numeral* and *temporal entities*. A well-performing NER system can help the downstream tasks of Machine Translation (Babych and Hartley, 2003), Information Extraction (Neudecker, 2016), and Questions Answering (Moldovan and Surdeanu, 2002). With the recent surge in the NER research (Sohrab and Miwa, 2018; Plank, 2019; Copara et al., 2020; Grancharova and Dalianis, 2021), the NLP community has also created large annotated datasets for the NER task (Ali et al., 2020; Ding et al., 2021) including code-mixed datasets (Singh et al., 2018). Research in NER has seen remarkable progress since the early approaches and evaluation metrics proposed by Sang (2002; Sang and De Meulder (2003). The task of NER belongs to the class of NLP problems, which can be modelled as a 'sequence labelling' problem akin to the tasks of Part-of-Speech (PoS) tagging and chunking. With the advent of deep learning-based approaches, sequence labelling tasks have invited much attention with successful methods like BiLSTM-CRF (Huang et al., 2015) and Transformers architecture-based fine-tuning (Vaswani et al., 2017; Wolf et al., 2019). However, these methods require significant data to produce a well-performing NER system for any language.

	Ours	Wiki ANN	Fire 2014	IJCNLP 2008
Sentences	109146	7000	9622	21833
Tokens	2220856	41256	116103	541682
Person	37605	22959	2112	4235
Location	198282	20131	2268	4307
Organization	26509	14204	170	1272

Table 1: Comparison of HiNER data statistics with existing Hindi NER datasets

NLP for Indian languages has shown progress with the availability of large language models (Kumar et al., 2020; Kakwani et al., 2020; Khanuja et al., 2021) which can help perform various NLP tasks. However, there has been little progress in terms of producing NER datasets for Indian languages, especially for Hindi, which approximately 342 million people speak across the world[1]. NER systems trained on our dataset are expected to perform better than the existing systems trained on lesser data. Existing datasets are either much smaller or have been automatically annotated (silver standard), rendering them incapable of performing the NER task with high accuracy. Moreover, during the creation of a Hindi NER system, one faces various linguistic challenges like:

[1]Wikipedia: List of Language by Speakers

No Capitalization: Unlike English or other languages which use the Latin script, Hindi does not have capitalization as a feature which should have been helpful for performing the NER task,

Ambiguity: Proper nouns in Hindi can be ambiguous as the same word can belong to a different PoS category. For example, a common Indian female name like '*Pushpa*' can be both a proper noun and a common noun meaning 'flower',

Spelling Variations: The spelling of some words in Hindi can differ depending on the local region in India. For example, the concept or sense of 'Plant' can be denoted by both the words- '*vanaspati*' and '*banaspati*',

Free Word Order: Languages like Hindi, which follow a free word order, make the NER task more challenging as computational approaches can not be complemented with a pattern of PoS tags, or strict word order.

Due to the challenges discussed above, it is imperative to train Hindi NER models with a sizeable human-annotated dataset so that deep learning-based approaches can generalize and perform well.
This paper describes our longstanding efforts toward creating a sizeable human-annotated dataset for Hindi NER, which we call "HiNER". We collect this dataset with the help of one annotator and perform experiments to evaluate the efficacy of various deep learning-based approaches. We also include the current public datasets in these experiments and compare the performance of these approaches across datasets. Our work also describes the NER tool developed in-house to help our annotators. This tool also provides a NER service on the back-end, which helps tag the NER data initially, and allows our annotators to post-edit the NER tags with ease. We describe the creation of the back-end NER engine in detail. We also discuss our dataset regarding the various sources and domains and provide an in-depth analysis of the NER tag-set we use for our dataset. The contributions of this work are summarized below:

- We collect a large manually annotated NER dataset for Hindi (HiNER) and release it publicly.
- We evaluate the performance of various deep learning-based NER approaches on our dataset and compare the performance with other publicly available datasets.
- We also release our data, code and models.

2. Related Work

For the task of Named Entity Recognition, much pre-existing literature attempts to solve the problem in different languages and domains. However, in this section, we discuss existing literature for Hindi and other Indian languages. We also describe research that highlights different approaches for the NER task. The IJCNLP 2008 NER dataset comprises NER data in five languages, namely Hindi, Bengali, Oriya, Telugu, and Urdu (IJCNLP, 2008). This data has been used extensively in previous research for the Hindi NER task (Ekbal et al., 2008; Gupta and Bhattacharyya, 2010; Bhagavatula et al., 2012; Gali et al., 2008; Saha et al., 2008b; Saha et al., 2008a). The FIRE 2014 dataset (Lalitha Devi et al., 2014) consists of NER data in four languages, namely Hindi, Tamil, Malayalam, and English (Choudhury et al., 2014). Similarly, the WikiANN data (Pan et al., 2017) consists of NER data in 282 languages, including Hindi; however, it is tagged automatically and a known 'silver-standard' dataset for the NER task. Moreover, it consists of only 10000 sentences in total. Rahimi et al. (2019) utilise transfer learning for multilingual NER and discuss their results for 41 languages in zero-shot, few-shot and high-resource scenarios. Singh et al. (2018) use Long Short Term Memory (LSTM), Decision Trees, and Conditional Random Fields (CRF) to perform the NER task on code-mixed Hindi-English social media text. Past research has also tried to utilise voting algorithm-based hybrid approaches, which take CRF, Maximum Entropy (MaxEnt) and rules into account (Srivastava et al., 2011). The authors use the IJCNLP-08 dataset for Hindi, and their approach achieved 82.95 as the F-score. Gupta and Bhattacharyya (2010) also identify a local context within the global information for the task of Hindi NER and report a performance gain of about 10% resulting in a 72% F1 score.

Recent work on Indian language NER utilises various deep learning-based approaches for the task. Singh et al. (2021) utilise a Bidirectional LSTM (BiLSTM) architecture with the help of contextualized ELMo word representations (Peters et al., 2018). Similarly, for the Hindi NER task, Athavale et al. (2016) explore the use of BiLSTM and utilise multiple datasets to report around 77.48% F1 score for all tags. Among multilingual approaches, past research has attempted to utilise morphological and phonological sub-word representations to help the NER task for four languages, including Hindi (Chaudhary et al., 2018). C S and Lalitha Devi (2020) also proposes various typological features and proposes a machine learning-based approach for the NER task in many language families. (Murthy et al., 2018a; Murthy et al., 2018b) demonstrate on FIRE 2014 data that training with combined labelled data of multiple languages can help in Indian language NER. With the help of non-speaker annotations, Tsygankova et al. (2020) show that even without the help of native speakers of the language, manual annotation for a NER task helps perform better than the available cross-lingual methods, which use modern contextualised representations. Focusing on the challenge of code-switching in NER data, Aguilar et al. (2020) propose a new benchmark for code-

Data Splits	#sentences	#words	Split Size
Training	76025	1382979	70%
Development	10861	200259	10%
Testing	21722	553961	20%
Total	108,608	2,137,199	100%

Table 2: **HiNER** dataset statistics in terms of the number of sentences (#sentences), number of words (#words), and the splits created for the Hindi NER task

switching they call LinCE and perform experiments for Hindi-English code-switched data to show encouraging results (75.96% F1). As discussed earlier, there is past research on NER, including NER for the Hindi language, but there are not sufficiently large datasets for the task of Hindi NER. With this paper, we release a large NER dataset collected **over several years with the help of a single annotator** and show its efficacy with the use of various available language models for Indian languages.

3. Dataset Creation

In this section, we discuss the creation of our dataset in detail. We follow the same guidelines as the CoNLL-2003 NER Shared task (Tjong Kim Sang and De Meulder, 2003). The CoNLL-2003 NER Shared task[2] contains the following tags: *Person*, *Location*, *Organization*, and *MISC*. While the CoNLL-2003 data does not contain *TIMEX* and *NUMEX* tags, these tags are part of Onto-notes (Pradhan et al., 2013) and we add *TIMEX* and *NUMEX* tags as part of our tag-set. During the annotation process, we observed the *MISC* tag to be too coarse and further include the *Language*, *Game*, *Literature*, *Religion*, and *Festival* as separate tag entities. We believe these fine-grained tags help create a more detailed NER dataset, thus helping the computational models be more accurate with the NER task. Some of challenges in Hindi NER is there is no capitalization, no use of camel case for names, and the free word order. In the following sub-section, we discuss the statistics of this dataset in detail. As one annotator has annotated the data, we can not provide any inter-annotator agreement details with our paper.

3.1. Dataset Statistics

We annotate data from the ILCI Tourism domain (Jha, 2010) and a subset of the 'news' domain corpus from Goldhahn et al. (2012). We take a subset of 9989 sentences out of 25,000 sentences from the ILCI tourism domain and the rest from the news domain. The dataset includes a total of 108,608 sentences. The number of entities for each tag is shown in Table 3 for a total of 11 tags. We also show the statistics of FIRE 2014 dataset in Table 1 for comparison.

[2] https://www.clips.uantwerpen.be/conll2003/ner/annotation.txt

	Train	Dev	Test	Total
PERSON	26310	3771	7524	37605
LOCATION	137995	20100	40187	198282
NUMEX	17194	2555	4662	24411
ORGANIZATION	18508	2645	5356	26509
MISC	4070	553	1080	5703
LANGUAGE	4187	571	1190	5948
GAME	1214	180	369	1763
TIMEX	13047	1762	3653	18462
RELIGION	823	133	234	1190
LITERATURE	597	74	181	852
FESTIVAL	203	30	40	273
Total	224148	32374	64476	320998

Table 3: Number of Entity mentions (Phrases) in *Train*, *Dev*, *Test* splits for the **HiNER dataset**

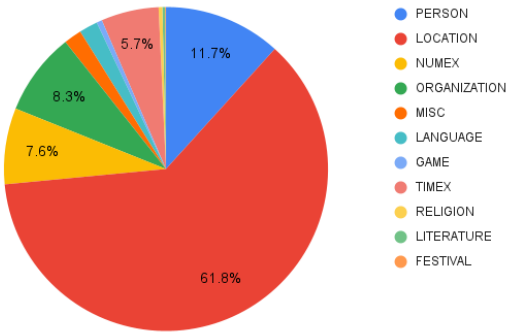

Figure 1: HiNER Tagset Details

3.2. NER Tool

To ease the annotation task, we create an online tool based on PaCMan (Kanojia et al., 2014). We modify the architecture of PaCMan to allow the upload of untagged NER data [3]. Further, we make changes in the tool front-end to show the full tag-set on the source and target sides of the screen as shown in Figure 2. The untagged data is also shown on the left side of the screen in a text box for clarity to the annotator; however, the annotator must tag the sentence on the right side. Borrowing a feature from the PaCMan interface, we modify the customized right-click-based context menu for different NER tags. The annotator must go through the sentence manually, highlight the named entity and then right-click to provide it with the correct label. This simplified annotation process allows our annotators to label the data with ease. The tool stores the data on a MySQL-based back-end and allows for downloading data files from the interface. Each time an annotator

[3] Link: Tool Interface

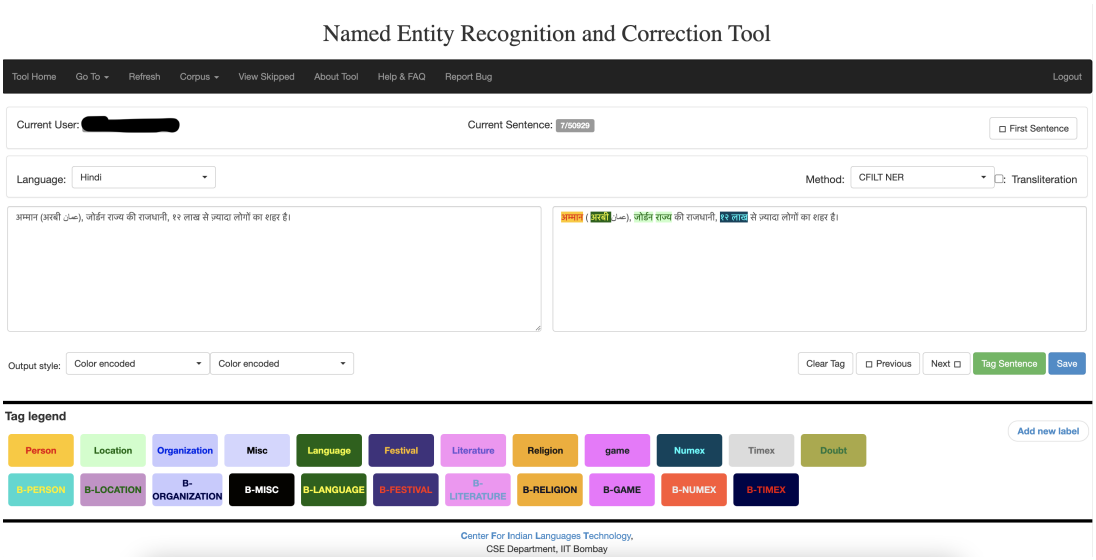

Figure 2: NER Tool Screenshot showing the tool interface with redacted user name to preserve anonymity.

Sentence	Confusion and Resolution
उच्च न्यायालय राज्य की सर्वोपरि न्यायिक सत्ता है ,. . . **uchcha nyAyAlaya** rAjya kI sarvopari nyAyika sattA hai , **high court** state's supreme judicial authority is **High Court** is the supreme judicial authority of the state.	Will *high court* be tagged as Organization or not? Every state in India has a high court. Since high court doesn't refer to a particular high court as in *Bombay High Court*, we leave it untagged.
चित्र छाया हिन्दी की एक फिल्मी मासिक पत्रिका है। **chitra ChAyA** hindI kI eka philmI mAsika patrikA hai chitra ChAyA hindi's one filmy monthly periodical is **Chitra Chhaya** is a Hindi film monthly magazine.	Should *chitra ChAyA* be tagged as *Organization* or *Literature*? In the context provided, *chitra ChAyA* refers to the periodical and not the organization, we tag them as *Literature*
विष्णु के वाहन **गरुड** हैं। viShNu ke vAhana *garuDa* haiM viShNu's vehicle *garuDa* is **Garuda** is the vehicle of Vishnu.	Will *garuDa* be tagged as *Person* name or not a named entity? Here, *garuDa* is the name of an eagle from Hindu mythology. According to CoNLL 2003 entity guidelines, it should be tagged as *Person* entity
रिक्की जय (सम्राज जयमंगल) ट्रिनिडाड के **चटनी संगीत** का कलाकार है। rikkI jaya (samrAja jayamaMgala) TriniDADa ke **chaTanI saMgIta** kA kalAkAra hai rikkI jaya (samrAja jayamaMgala) Trinidad's *chaTanI saMgIta* artist is Rikki Jai (Emperor Jayamangala) is a Trinidadian **chutney music** artist.	What should be the entity label of *chaTanI saMgIta* ? Here *chaTanI saMgIta* refers to a fusion genre of Indian folk music, specifically Bhojpuri folk music, with local Caribbean calypso and soca music, and later on Bollywood music. Hence, will be tagged as *Misc* entity.

Table 4: Sentences flagged by the annotator with the entity highlighted and the reasoning for the final decision.

progresses onto the following sentence, the previously tagged sentence is saved automatically. The tool also saves the annotation state in the database, thus allowing an annotator to arrive at the next untagged instance in the database when they log on later. We further simplify the annotation by providing them with a baseline NER engine that allows them to tag the sentence initially and simply "post-edit" the annotations and save the correctly labelled sentence. We describe this baseline NER engine in the following subsection.

3.3. NER Engine

We developed a NER engine to provide Named Entity suggestions to our annotators. Each sentence from our dataset is presented on the tool interface as shown in the screenshot (Figure 2), and a button ("Tag Sentence") which allows the NER engine to perform NE tagging of the sentence on the back-end. The tagged sentence is shown to the annotator on the annotation screen's right side, which can be edited later. Our annotators reported that they could easily modify the tool's engine

errors. This NER engine was developed using FIRE 2013 Hindi NER corpus (RK and Lalitha Devi, 2013). Due to the limited size of the training corpus, it was hard to create a tagger that could learn a generic sequence of tags. To support the model, we employed word2vec (Mikolov et al., 2013) to learn the semantic embeddings for single and multi-word tokens based on a large Hindi Wikipedia dump. These learned embeddings were then used to train a simple perceptron-based neural network model to infer named entities. A separate service was created in conjunction with the front-end UI of our NER tool to handle the annotation requests. Our annotators reported that this engine was prone to errors, especially when tagging multi-word named entities, but it could handle commonly used named entities.

3.4. Annotation Ambiguity

As only one annotator annotated the data, ensuring that the dataset's quality is not compromised is essential. We encouraged the annotator to raise reports for entities he was not confident in tagging. The authors then take a majority voting on such instances to assign an appropriate entity or not an entity label. We now provide a few examples of such instances raised by the annotator in Table 4.

4. Dataset Evaluation

In this section, we discuss the evaluation of our dataset based on different approaches to NER. With the help of our annotator, we collected the NER-labelled dataset as described above. We perform the task of Hindi NER with the help of various contextual language models and in different settings. With our dataset, we create a data split of 70% for training, 10% for development, and 20% for testing, with statistics, as shown in Table 2. We ensured a balanced percentage of tags in each of the splits with stratification, as can be seen from Table 3.

4.1. Experimental Setup

With the advent of contextualized word representations, various language models have been proposed which can be utilized to perform NLP tasks (Devlin et al., 2019; Conneau et al., 2020; Kakwani et al., 2020; Khanuja et al., 2021). We use these four models to evaluate the performance of the NER task on our dataset. Additionally, we use the FIRE 2014 dataset to compare the efficacy of both datasets and present the results in the next section. We also utilize the models trained on our data and test on the FIRE 2014 test split to evaluate the model performance in a cross-dataset scenario.

We use the variation mBERT$_{base-cased}$ of multilingual BERT (mBERT), which supports 104 languages, and has 12 layers with 768 hidden layers, along with a total of 110M parameters. We use *XLM-R$_{base}$* and *XLM-R$_{large}$* (Conneau et al., 2020) which are pre-trained multilingual language models to fine-tune for NER task. However, IndicBERT (Kakwani et al., 2020), and MuRIL (Khanuja et al., 2021) are more suited to the task as it supports Indian languages in particular and is trained on shared vocabulary from Indic languages. IndicBERT is trained on 12 major Indian languages, including Hindi, is trained on around 9 billion tokens, and has a restriction on the maximum sequence length (128). Similarly, MuRIL is a model pre-trained on 17 Indian languages and their transliterated counterparts.

We perform hyper-parameter tuning of each model and select the hyper-parameters giving the best F-Score on the development set. The model is trained using the best hyper-parameter for 5 runs. We report the mean

	Indic-BERT	**mBERT**	**MuRIL**	**XLM-R$_{base}$**	**XLM-R$_{large}$**
Festival	9.52 ± 11.90	8.57 ± 17.14	0.00 ± 0.00	11.34 ± 14.53	**46.73 ± 23.96**
Game	50.05 ± 8.33	50.92 ± 20.52	40.88 ± 22.96	47.57 ± 10.63	**59.63 ± 7.94**
Language	89.22 ± 1.15	90.07 ± 1.13	90.08 ± 1.02	90.64 ± 0.56	**91.42 ± 0.57**
Literature	21.64 ± 26.12	53.56 ± 10.93	44.23 ± 22.17	40.54 ± 23.39	**56.69 ± 6.32**
Location	94.10 ± 0.56	93.92 ± 0.57	94.81 ± 0.37	94.07 ± 0.76	**94.86 ± 0.40**
Misc	56.14 ± 10.97	61.24 ± 10.99	62.84 ± 4.22	60.38 ± 12.19	**67.86 ± 2.19**
NUMEX	65.56 ± 3.25	67.21 ± 1.50	68.31 ± 1.77	66.72 ± 2.32	**69.10 ± 0.95**
Organization	76.68 ± 1.33	74.81 ± 3.10	78.26 ± 2.46	76.02 ± 2.73	**78.76 ± 1.70**
Person	83.65 ± 0.50	81.10 ± 1.70	84.60 ± 1.30	83.04 ± 0.86	**85.14 ± 0.94**
Religion	65.94 ± 3.20	68.55 ± 7.58	53.43 ± 26.74	67.70 ± 5.78	**72.27 ± 2.68**
TIMEX	80.20 ± 1.11	81.15 ± 1.24	81.17 ± 1.20	79.50 ± 0.85	**80.63 ± 1.05**
Micro	87.44 ± 0.62	87.11 ± 1.01	88.27 ± 0.92	87.36 ± 1.09	**88.73 ± 0.60**
Macro	62.97 ± 5.19	66.46 ± 5.93	63.51 ± 7.36	65.23 ± 6.04	**73.01 ± 3.35**
Weighted	87.25 ± 0.88	87.06 ± 1.28	88.27 ± 1.08	87.29 ± 1.23	**88.78 ± 0.57**

Table 5: Test Set F1-Score of various pre-trained LMs on our HiNER dataset. This table reports a mean F1-score and its standard deviation over 5 runs.

	Indic-BERT	mBERT	MuRIL	XLM-R$_{base}$	XLM-R$_{large}$
Location	94.33 ± 0.63	94.44 ± 0.24	94.95 ± 0.25	95.07 ± 0.20	95.06 ± 0.33
Organization	78.29 ± 1.57	78.42 ± 1.13	79.87 ± 0.81	79.57 ± 0.62	80.53 ± 0.40
Person	84.70 ± 0.61	82.20 ± 0.98	85.66 ± 0.48	85.18 ± 0.66	85.34 ± 0.54
micro avg	91.37 ± 0.67	91.10 ± 0.34	92.09 ± 0.27	92.06 ± 0.27	92.20 ± 0.22
macro avg	85.77 ± 0.91	85.02 ± 0.66	86.83 ± 0.41	86.61 ± 0.40	86.98 ± 0.22
weighted avg	91.34 ± 0.71	91.08 ± 0.35	92.11 ± 0.29	92.11 ± 0.27	**92.22 ± 0.22**

Table 6: Test Set F1-Score of various pre-trained LMs on our HiNER dataset (Collapsed). This table reports a mean F1-score and its standard deviation over 5 runs.

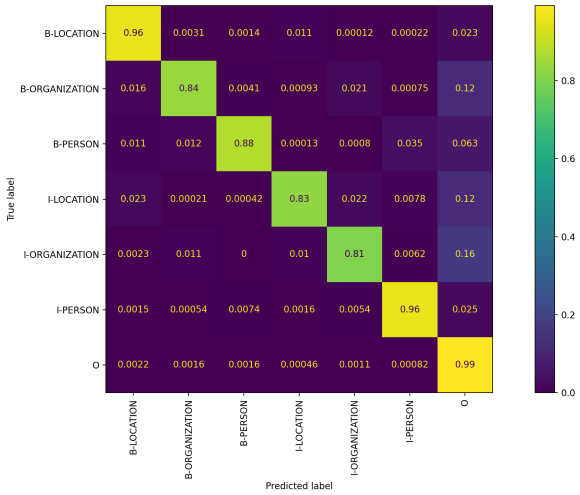

Figure 3: Confusion Matrix on HiNER Collapsed data from XLM-R$_{large}$ Model

and standard deviation over all runs in all our experiments. Considering the batch size and learning rate as hyper-parameters, we provide the following variations for batch size $\{8, 16, 32\}$. Due to GPU memory limitations, we did not experiment with larger batch sizes. Similarly, we vary the learning rate in the following range {1e-3, 3e-3, 5e-3, 1e-4, 3e-4, 5e-4, 1e-5, 3e-5, 5e-5, 1e-6, 3e-6, 5e-6 } and select the best learning rate. We perform experiments on two variations of our dataset: one with all 11 tags, and the collapsed version with only 3 tags (*Person*, *Location*, *Organization* tags). For FIRE 2014 data (Lalitha Devi et al., 2014), similar to Murthy et al. (2018a; Murthy et al. (2018b), we consider only *Person*, *Location*, *Organization* tags. We use the I-O-B encoding as input format for model training and report the results using Seqeval (Nakayama, 2018) generate evaluation statistics.

4.2. Evaluation Results

From Table 5, we observe that *XLM-R$_{large}$* performs the best on our dataset followed by *MuRIL*. *Festival* entity has the lowest performance across all models and also has highest standard deviation across different runs compared to other tags. All the models are able to identify *Language*, *Location*, *Person*, and, *TIMEX* entities with relatively high accuracy.

Model	HiNER (collapsed)	FIRE 2014	HiNER → FIRE Zero-Shot
IndicBERT	91.37 ± 0.67	62.79 ± 0.68	46.82 ± 1.87
mBERT	91.10 ± 0.34	62.14 ± 0.59	47.60 ± 1.62
MuRIL	92.09 ± 0.27	62.58 ± 2.44	55.26 ± 1.59
XLM-R$_{base}$	92.06 ± 0.27	65.63 ± 0.76	49.48 ± 0.79
XLM-R$_{large}$	**92.20 ± 0.22**	**66.75 ± 0.30**	**49.52 ± 3.12**

Table 7: Test Set Micro F1-Score of various pre-trained LMs on both datasets where HiNER (collapsed) is our dataset with only the *Person*, *Location* and *Organization* tags. This table reports a mean F1-score and its standard deviation over 5 runs.

Table 6 reports results on our collapsed data. We focus on *Person*, *Location*, and, *Organization* tags here. Similar to our earlier results, *XLM-R$_{large}$* performs the best followed by *MuRIL* model. Unsurprisingly, *Organization* entity has relatively lowest F1 score compared to *Person* and *Location* entities.

Zero-Shot Performance Test

Table 7 reports the results from our experiments in the Zero-Shot experiment setup. For this set of experiments, we use FIRE 2014 Hindi NER data (Lalitha Devi et al., 2014). When we perform a zero-shot transfer from our dataset to the FIRE 2014 dataset, the results are poor compared to training on the FIRE 2014 dataset. Surprisingly, *MuRIL* model performs the best in the zero-shot transfer set-up compared to other pre-trained language models.

5. Discussion

We plot the confusion matrix on HiNER Collapsed data from one of the runs of XLM-R$_{large}$ Model in Figure 3. We observe that majority of the errors involve tagging a named entity as not a named entity followed by boundary errors *i.e.,* mislabelling *B-Person* with *I-Person* and vice-versa. Also *Organization* entities tends to be confused with *Location* entities. The majority of the errors are produced when the model cannot identify a token as a name itself. Additionally, we report a detailed analysis of the types of errors made

by the XLM-R_{large} Model on HiNER data. Table 10 provides more detailed insights into the performance of the system by reporting *strict*, *exact* evaluation metrics (Chinchor and Sundheim, 1993). We use the *nervaluate* package[4] to calculate the above statistics for each entity type. Specifically, we pick the predictions from one of the runs using *XLM-R_{large}* as this model consistently gave better results compared to the other pretrained language models. We use two different evaluation schemas mentioned in the Table 8. *Exact* encourages models to identify the named entity phrase correctly while ignoring the type mismatch.

We observe that for some entity type like *Location, NUMEX, Organization Missed* errors are more than the *Spurious* errors. On the other hand, for entity types like *Person, Misc, Language, Game, TIMEX Spurious* errors are more. We additionally report F1-Score according to the evaluation schema for each entity type. The most challenging entity categories are *Literature*, *Festival*, *MISC*, *Language*, *Religion*, and *Game* entities. We observe that the model is able to identify *Misc, Language, Religion, Literature* as named entities but unable to assign the correct entity type. This cab seen in the F-Score different between *Strict* and *Exact* evaluation schema.

Evaluation Schema	Explanation
Strict	The exact boundary surface string match and entity type match
Exact	The exact boundary match over the surface string, regardless of the type

Table 8: Short Description of the Evaluation Schema used

For each type of evaluation schema (*i.e., strict and exact*) we report the following categories of errors listed in Table 9.

Error type	Explanation
Correct	The gold annotations and the system predictions are the same
Incorrect	The system prediction and the gold annotation don't match
Missing	The system prediction classifies an entity as not a named entity
Spurius	The system prediction classifies a non named-entity as a named entity

Table 9: Short Description of the Categories of Errors

	Error Category	Strict	Exact
Person	Correct	6475	6565
	Incorrect	622	532
	Missed	427	427
	Spurious	537	537
	F1	0.8543	0.8662
Location	Correct	37960	38113
	Incorrect	1028	875
	Missed	1199	1199
	Spurious	688	688
	F1	0.9506	0.9545
NUMEX	Correct	3047	3097
	Incorrect	567	517
	Missed	1048	1048
	Spurious	526	526
	F1	0.6923	0.7037
Organization	Correct	4195	4263
	Incorrect	535	467
	Missed	626	626
	Spurious	544	544
	F1	0.7893	0.8021
Misc	Correct	804	882
	Incorrect	137	59
	Missed	139	139
	Spurious	265	265
	F1	0.7034	0.7717
Language	Correct	1115	1133
	Incorrect	60	42
	Missed	15	15
	Spurious	42	42
	F1	0.9265	0.9414
Game	Correct	276	279
	Incorrect	57	54
	Missed	36	36
	Spurious	145	145
	F1	0.6517	0.6588
TIMEX	Correct	3018	3055
	Incorrect	328	291
	Missed	307	307
	Spurious	363	363
	F1	0.8199	0.8299
Religion	Correct	175	183
	Incorrect	26	18
	Missed	33	33
	Spurious	32	32
	F1	0.7495	0.7837

[4] https://github.com/MantisAI/nervaluate

Table 10, continued

		Strict	Exact
Literature	Correct	104	116
	Incorrect	35	23
	Missed	42	42
	Spurious	21	21
	F1	0.6100	0.6804
Festival	Correct	26	26
	Incorrect	6	6
	Missed	8	8
	Spurious	15	15
	F1	0.5977	0.5977

Table 10: Detailed Strict and Exact Results on HiNER data from XLM-R_{large} Model

6. Conclusion and Future Work

We describe our efforts to create a sizeable human-annotated dataset, HiNER, for the task of Named Entity Recognition in the Hindi language. We discuss the motivation for this research, the challenges specific to Hindi NER, and provide coverage of the past research performed for the NER task in Hindi. We discuss the dataset creation in detail and provide an in-depth analysis of the tag-set used to label our NER data. We also describe the NER annotation tool created to help our annotators along with the NER engine it utilises to label the data initially on the tool interface. We split our data and performed experiments to evaluate different language models to perform the NER task by fine-tuning them. We also perform similar experiments on another dataset for a comparative evaluation. We discuss our results in detail and show how large human-annotated NER data is essential for the task of Hindi NER. We release this dataset and the models we train; for the NLP community to utilise them for the downstream NLP tasks. We choose the CC-BY-SA 4.0 Licensing terms to release this data. In future, we plan to keep extending this dataset with the help of our ongoing annotation process.

Bibliographical References

Aguilar, G., Kar, S., and Solorio, T. (2020). LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France, May. European Language Resources Association.

Ali, W., Lu, J., and Xu, Z. (2020). SiNER: A large dataset for Sindhi named entity recognition. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2953–2961, Marseille, France, May. European Language Resources Association.

Athavale, V., Bharadwaj, S., Pamecha, M., Prabhu, A., and Shrivastava, M. (2016). Towards deep learning in Hindi NER: An approach to tackle the labelled data sparsity. In *Proceedings of the 13th International Conference on Natural Language Processing*, pages 154–160, Varanasi, India, December. NLP Association of India.

Babych, B. and Hartley, A. (2003). Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003*.

Bhagavatula, M., Santosh, G., and Varma, V. (2012). Language independent named entity identification using wikipedia. In *Proceedings of the First Workshop on Multilingual Modeling*, pages 11–17.

C S, M. and Lalitha Devi, S. (2020). A deeper study on features for named entity recognition. In *Proceedings of the WILDRE5– 5th Workshop on Indian Language Data: Resources and Evaluation*, pages 66–72, Marseille, France, May. European Language Resources Association (ELRA).

Chaudhary, A., Zhou, C., Levin, L., Neubig, G., Mortensen, D. R., and Carbonell, J. G. (2018). Adapting word embeddings to new languages with morphological and phonological subword representations. *arXiv preprint arXiv:1808.09500*.

Chinchor, N. and Sundheim, B. (1993). MUC-5 evaluation metrics. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*.

Choudhury, M., Chittaranjan, G., Gupta, P., and Das, A. (2014). Overview of fire 2014 track on translitterated search. *Proceedings of FIRE*, pages 68–89.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (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, July. Association for Computational Linguistics.

Copara, J., Knafou, J., Naderi, N., Moro, C., Ruch, P., and Teodoro, D. (2020). Contextualized French language models for biomedical named entity recognition. In *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Atelier DÉfi Fouille de Textes*, pages 36–48, Nancy, France, 6. ATALA et AFCP.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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 Computa-*

tional Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Ding, N., Xu, G., Chen, Y., Wang, X., Han, X., Xie, P., Zheng, H.-T., and Liu, Z. (2021). Few-nerd: A few-shot named entity recognition dataset. *arXiv preprint arXiv:2105.07464*.

Ekbal, A., Haque, R., Das, A., Poka, V., and Bandyopadhyay, S. (2008). Language independent named entity recognition in indian languages. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*.

Gali, K., Surana, H., Vaidya, A., Shishtla, P. M., and Sharma, D. M. (2008). Aggregating machine learning and rule based heuristics for named entity recognition. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*.

Goldhahn, D., Eckart, T., and Quasthoff, U. (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, May. European Language Resources Association (ELRA).

Grancharova, M. and Dalianis, H. (2021). Applying and sharing pre-trained BERT-models for named entity recognition and classification in Swedish electronic patient records. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 231–239, Reykjavik, Iceland (Online), May 31–2 June. Linköping University Electronic Press, Sweden.

Gupta, S. and Bhattacharyya, P. (2010). Think globally, apply locally: using distributional characteristics for hindi named entity identification. In *Proceedings of the 2010 Named Entities Workshop*, pages 116–125.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

IJCNLP. (2008). IJCNLP NER Dataset. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*.

Jha, G. N. (2010). The TDIL Program and the Indian Language Corpora Intitiative (ILCI). In *Proceedings of the Seventh conference on International Language Resources and Evaluation, LREC 2010*.

Kakwani, D., Kunchukuttan, A., Golla, S., Gokul, N., Bhattacharyya, A., Khapra, M. M., and Kumar, P. (2020). inlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4948–4961.

Kanojia, D., Shrivastava, M., Dabre, R., and Bhattacharyya, P. (2014). PaCMan : Parallel corpus management workbench. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 162–166, Goa, India, December. NLP Association of India.

Khanuja, S., Bansal, D., Mehtani, S., Khosla, S., Dey, A., Gopalan, B., Margam, D. K., Aggarwal, P., Nagipogu, R. T., Dave, S., et al. (2021). Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Kumar, S., Kumar, S., Kanojia, D., and Bhattacharyya, P. (2020). "A Passage to India": Pre-trained word embeddings for Indian languages. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 352–357, Marseille, France, May. European Language Resources association.

Lalitha Devi, S., RK Rao, P., C.S, M., and Sundar Ram, R. V. (2014). Indian Language NER Annotated FIRE 2014 Corpus (FIRE 2014 NER Corpus). In *In Named-Entity Recognition Indian Languages FIRE 2014 Evaluation Track*.

Ma, X. and Hovy, E. H. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. *CoRR*, abs/1603.01354.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Moldovan, D. and Surdeanu, M. (2002). On the role of information retrieval and information extraction in question answering systems. In *International Summer School on Information Extraction*, pages 129–147. Springer.

Murthy, R., Khapra, M. M., and Bhattacharyya, P. (2018a). Improving NER Tagging Performance in Low-Resource Languages via Multilingual Learning. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 18(2), dec.

Murthy, R., Kunchukuttan, A., and Bhattacharyya, P. (2018b). Judicious selection of training data in assisting language for multilingual neural NER. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 401–406, Melbourne, Australia, July. Association for Computational Linguistics.

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

Neudecker, C. (2016). An open corpus for named entity recognition in historic newspapers. In *Proceedings of the Tenth International Conference on Lan-*

guage Resources and Evaluation (LREC'16), pages 4348–4352.

Pan, X., Zhang, B., May, J., Nothman, J., Knight, K., and Ji, H. (2017). Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Plank, B. (2019). Neural cross-lingual transfer and limited annotated data for named entity recognition in Danish. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 370–375, Turku, Finland, September–October. Linköping University Electronic Press.

Pradhan, S., Moschitti, A., Xue, N., Ng, H. T., Björkelund, A., Uryupina, O., Zhang, Y., and Zhong, Z. (2013). Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria, August. Association for Computational Linguistics.

Rahimi, A., Li, Y., and Cohn, T. (2019). Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy, July. Association for Computational Linguistics.

RK, P. and Lalitha Devi, S. (2013). NERIL: Named Entity Recognition for Indian Languages @ FIRE 2013–An Overview. In *FIRE 2013*.

Saha, S. K., Mitra, P., and Sarkar, S. (2008a). Word clustering and word selection based feature reduction for MaxEnt based Hindi NER. In *Proceedings of ACL-08: HLT*, pages 488–495, Columbus, Ohio, June. Association for Computational Linguistics.

Saha, S. K., Sarkar, S., and Mitra, P. (2008b). A hybrid feature set based maximum entropy Hindi named entity recognition. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Sang, E. T. K. (2002). Memory-based named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Singh, V., Vijay, D., Akhtar, S. S., and Shrivastava, M. (2018). Named entity recognition for Hindi-English code-mixed social media text. In *Proceedings of the Seventh Named Entities Workshop*, pages 27–35, Melbourne, Australia, July. Association for Computational Linguistics.

Singh, S., Patel, S., Shah, Y., Nargunde, R., and Ramteke, J. (2021). Context-based deep learning approach for named entity recognition in hindi. In *2021 International Conference on Communication information and Computing Technology (ICCICT)*, pages 1–5.

Sohrab, M. G. and Miwa, M. (2018). Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium, October-November. Association for Computational Linguistics.

Srivastava, S., Sanglikar, M., and Kothari, D. (2011). Named entity recognition system for hindi language: a hybrid approach. *International Journal of Computational Linguistics (IJCL)*, 2(1):10–23.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03. Association for Computational Linguistics.

Tsygankova, T., Marini, F., Mayhew, S., and Roth, D. (2020). Building low-resource ner models using non-speaker annotation. *arXiv preprint arXiv:2006.09627*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.