Development of a Low-Cost Named Entity Recognition System for Odia Language using Deep Active Learning

Tusarkanta Dalai NIT Rourklea tusarkantadalai@gmail.com
Tapas Kumar Mishra NIT Rourkela mishrat@nitrkl.ac.in
Pankaj K Sa NIT Rourkela PankajKSa@nitrkl.ac.in
Prithviraj Mohanty ITER, SOA University prithvirajmohanty@soa.ac.in
Chittaranjan Swain IIITM Gwalior cswain@iiitm.ac.in
Ajit Nayak ITER, SOA University ajitnayak@soa.ac.in

Abstract

Named Entity Recognition (NER) is a crucial component of Natural Language Processing (NLP) systems, which are utilized to extract significant information from massive quantities of unstructured textual data. The application of NER holds significant value in various NLP tasks, including but not limited to information retrieval, automatic question-answering systems, information extraction, and machine translation. Already, NER has accomplished fruitful achievements in English as well as in a number of other European languages. On the other hand, it is not yet well explored in Indian languages, primarily in the Odia language, remains insufficiently investigated owing to the absence of supporting tools and resources. In recent years, Machine Learning (ML) and deep learning (DL) based approaches have been able to achieve outstanding performance in constructing NLP tasks; nevertheless, these methods generally call for massive volumes of annotated corpus, which are costly to generate due to the need for domain-specific exports. Therefore, at present, researchers are utilizing the active learning approach, which involves the use of a sample selection technique in conjunction with supervised models. The aim of this approach is to minimize annotation expenses while optimizing the performance of ML and DL based models. The primary objective of this research is to develop a new active learning based NER system for Odia language.We applied a deep active learning (Deep-AL) strategy, and the deep active learning-based Odia NER system achieved nearly state-of-the-art performance. By utilizing only 38% of the original training data, we have achieved a maximum F1 score of 85.02%, which could save almost 62% of the cost for annotation.

1 Introduction

Identifying and classifying named entities (NE) into predefined classes such as person, location, organization, number, and time is a fundamental task in many NLP applications, commonly referred to as Named Entity Recognition (NER) (Nadeau and Sekine, 2007). Accurate identification of such entities is critical for extracting structured information from unstructured text. NER also plays a significant role in search engines for organizing, indexing, and linking named references consistently, thereby enhancing document searchability. For example, a NER system can help accurately determine individuals mentioned in news articles. Its utility has been well demonstrated in systems like Amazon Alexa and Apple Siri, particularly for Western languages. Moreover, NER is a key component in several downstream NLP tasks such as question answering, text summarization, machine translation (Bala Das et al., 2024, 2023; Das et al., 2025b,a), word-sense

Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025, pages 16–26, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-003

disambiguation, coreference resolution, and semantic search.

Despite significant progress in European and many Asian languages, NER remains underexplored in low-resource languages, particularly Odia. Literature indicates that very limited attention has been paid to Odia in the context of NLP tasks, including NER. While Indian languages have seen increasing research in computational linguistics, Odia continues to lack the necessary tools and resources. Thus, it becomes essential to investigate NER from the lens of Odia language processing. Though a few studies on Odia NER exist, many of the resources used are either not publicly accessible or lack proper documentation. This work builds on earlier research in developing POS taggers for Odia (Dalai et al., 2023, 2024), and represents a step forward in advancing sequence labeling tasks for the language.

The development of a robust Odia NER system is crucial for enhancing Odia NLP applications. Several approaches such as probabilistic methods, rule-based systems, deep learning models, and hybrid strategies have been employed to address NER tasks. These systems aim to automatically tag entities in text. However, existing Odia NER systems mostly rely on conventional approaches like rule-based or machine learning techniques. The lack of Odia linguistic resources, including grammar knowledge and handcrafted features, poses a significant challenge. Additional linguistic complexities such as free word order, no capitalization, high ambiguity, and morphological richness further complicate NER development in Odia.

To address these challenges, deep learning (DL) methods are being increasingly adopted in NER system development. Popular models like CNNs, RNNs, LSTMs, and GRUs have shown success in other languages by leveraging multiple neural network layers to extract higher-level features. However, DL-based models have not yet been effectively applied to Odia NER. This work aims to bridge that gap by evaluating DL-based approaches for Odia. Since DL models require large volumes of annotated data, which are expensive and time-consuming to produce, we adopt an active learning (AL) approach. AL combines sample selection strategies with supervised learning to reduce annotation costs while maintaining model effectiveness.

AL is especially useful in scenarios where collecting large labeled datasets is not practical. As a semi-supervised technique, AL focuses on reducing manual labeling during training by iteratively selecting the most informative samples. In this study, we develop an active learning-based NER system for Odia using a relatively small annotated corpus. We employ a subset of the Odia NER dataset (Dalai et al., 2025), comprising 10,950 sentences annotated across twelve entity classes. The dataset and methodology are described in subsequent sections.

2 Related Work

This section presents a comprehensive overview of research and development in Named Entity Recognition (NER). The earliest significant effort in this area was introduced by (Grishman and Sundheim, 1996) at the Sixth Message Understanding Conference (MUC-6) in 1996, where the NER task focused on identifying entities such as persons, organizations, locations, percentages, and currency. Following this, numerous researchers contributed to the growth of the field (Sang and De Meulder, 2003; Demartini et al., 2009; Balog et al., 2010). Several advancements were later made in Indian language NER through rule-based approaches (Gupta and Lehal, 2011; Alfred et al., 2014; Riaz, 2010; Sasidhar et al., 2011). While such systems often yielded strong results,

they had notable limitations, including high dependence on manual effort, slow learning capability, and substantial time requirements. Moreover, rule-based NER systems tend to be language-specific, making them difficult to adapt across different linguistic contexts. Due to these drawbacks, attention gradually shifted towards statistically-driven machine learning algorithms, which offered more flexibility and scalability for NER development.

The advancement of the NER system has encompassed the amalgamation of diverse statistical methodologies, such as the Support Vector Machine (SVM), Maximum Entropy (ME)(Saha et al., 2012), Hidden Markov Model (HMM)(Bikel et al., 1997; Morwal et al., 2012), Conditional Random Fields (CRF)(Mccallum, 2003), and other related techniques. These systems achieve this by integrating rule-based and ML-based approaches (Chopra et al., 2012; Biswas et al., 2010; Srivastava et al., 2011). Although machine learning-based NER systems exhibit remarkable performance, these systems nevertheless have a number of major disadvantages. These limitations include the requirement for extensive annotated datasets, the challenge of selecting an appropriate feature set, and the choice of an appropriate learning algorithm. Furthermore, researchers have initiated the development of DL models that avoid the need for traditional methodologies for the development of sequence labeling task.

Initially, (Collobert et al., 2011) devised an English NER model by utilizing characteristics acquired from word embeddings (WE) that were trained on an extensive collection of unlabeled data. (Chiu and Nichols, 2016) developed a NER system for the English language and this model incorporated both Bi-LSTM and CNN architectures to capture character-level details. In a similar manner, (Ma and Hovy, 2016) proposed a NER model for the English. The model, based on a combination of Bi-LSTM, CNN, and CRF, incorporates various deep learning techniques. In addition, (Athavale et al., 2016) developed a model for Hindi NER systems, which integrates pre-trained word embeddings with a Bi-LSTM architecture and a softmax layer. (Gupta et al., 2018) introduced an additional neural network model for NER that utilizes deep learning techniques. This model specifically focuses on code-mixed Indian social media content and employs a gated recurrent unit (GRU) along with character- and word-layer embeddings. However, the utilization of deep learning-based methods often necessitates a large quantity of annotated corpora. Nonetheless, the process of constructing such datasets demands a significant investment of time and extensive manual effort. Active learning has demonstrated promising results in situations where there is a limited corpus, thereby reducing the requirement for a large dataset. The system selects samples for labeling in an efficient manner. The active learning technique enables the algorithm to make informed decisions regarding the selection of instances for labeling, as opposed to the supervised learning mode, where a random subset of unlabeled instances is generated and labeled.

Many NLP applications, including information extraction (Settles and Craven, 2008), text classification (Tong and Koller, 2001), and word sense disambiguation (Zhu and Hovy, 2007), which need annotation from a huge pool of unannotated data to build a supervised ML model have benefited from Active Learning (AL) methodologies. However, the traditional AL algorithm fails to address high-dimensional data. Therefore, it is anticipated that the combination of active learning and deep learning will produce better results. Deep active learning has been employed extensively in numerous applications like text categorization (Schröder

and Niekler, 2020; Zhang et al., 2017), image recognition (Gal et al., 2017; Gudovskiy et al., 2020), visual question answering (Lin and Parikh, 2017), and object detection (Aghdam et al., 2019; Feng et al., 2019).

A handful of NER model construction initiatives have been proposed for the Odia language. (Das and Patnaik, 2013) proposed the first Odia NER system; it made use of a support vector machine and attained an F1 score of 80% by feeding the feature set as language-specific rules, gazetteers, and context patterns. Following this, (Das et al., 2015) introduced an Odia NER system based on ML and trained on a manually annotated corpus of 1,000 sentences. For the purpose of data labeling, a set of ten tags was considered. This NER system achieved an F1 score of 81%. Subsequently, (Balabantaray et al., 2013) developed a NER system for the Odia language that was based on CRF, and they acquired an F1 score of 71%. In order to evaluate the effectiveness of the NER task, a variety of feature sets were generated using gazetteers and POS tags, respectively.

Based on our review of the relevant literature, we found that researchers have not paid much attention to Odia for NLP tasks such as NER, and only a small amount of study has been conducted on the language. Deep learning-based strategies were not utilized to their full potential when building the Odia Natural Language Engineering (NER) system.

3 System Model

3.1 Active Learning

This subsection describes the algorithmic procedure of Active Learning (AL), as depicted in Figure 1. The initial training samples for a machine learning or deep learning-based model are annotated by domain experts, and it picked according to a predefined strategy. After that, the annotated data is used to train the model; the unlabeled samples are ranked using a predetermined set of rules, and the best n samples are selected for annotation. Next, the annotated data are added to the training set, and the model is retrained using the updated training data. Iteration is performed on both the learning process and the selecting process up until the termination condition is met. It is very clear that the AL process ought to address the three significant concerns. The first step is the production of the initial training set, the second is the selection of an appropriate method for sample selection, and the third is the effective setup of the iterative process and the quit condition.

In this research, we modeled a real-world AL framework on pool-based resources. Even though we pre-annotated every sentence in our corpus, we did not make use their labels until the query algorithms picked them out.

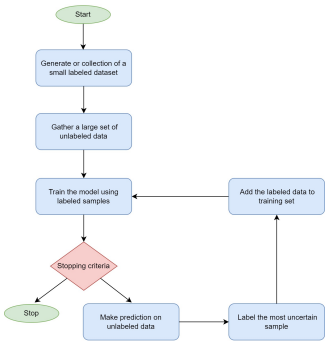

Figure 1: The process of active learning

Existing active learning methods have already demonstrated promising results in sequence labeling tasks. Three uncertainty-based strategies are implemented in our experiment: Least Confidence (LC), Bayesian Active Learning by Disagreements (BALD) and Maximum normalized log-probability (MNLP).

3.2 Deep Active Learning

This subsection presents a complete and methodical approach for the Deep active learn-

ing (Deep-AL)-based Odia NER system. DL has a high learning capacity when it comes to the processing of high-dimensional data and the automatic extraction of features, but AL has the ability to significantly minimize the costs associated with labeling the data. Consequently, it is clear to combine active learning and deep learning since this will considerably increase their applicability. Deep-AL was proposed by taking into account the combined benefits of the two methodologies. The framework of the Deep-AL model for the NER task of Odia language is illustrated in Figure 2. A deep learning model must first be initialized and pre-trained on labeled training data to extract features from unlabeled samples. After that, we chose samples by employing the corresponding query strategy, query the label by the manual annotator to construct a new training set, trained the deep learning model by making use of the updated training data, and then simultaneously updated the unlabeled pool. This method is repeated until the predetermined termination criteria are met. The Deep-AL architecture can, in its most basic form, be broken down into two parts: the AL query method applied to the unlabeled dataset and the DL model training procedure.

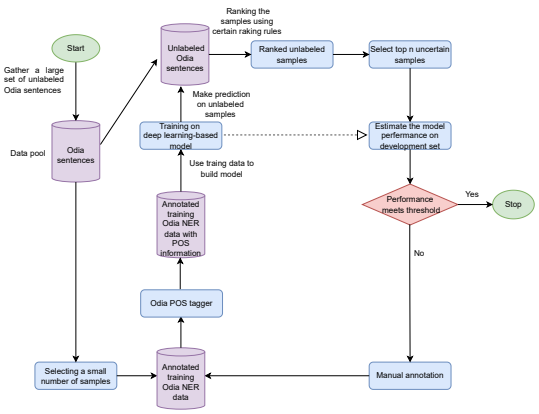

Figure 2: Framework of deep active learning model for Odia NER task

3.2.1 Deep Learning

The majority of AL approaches call for frequent retraining of the model when newly labeled instances are annotated. This is required in order to ensure optimal performance. As a consequence of this, it is essential that the model be capable of being retrained in a time-efficient manner. On the other hand, we would like to match with state-of-the-art deep learning-based models in terms of performance. In order to accomplish this, we must first determine the various deep learning architectures that comprise the Odia NER system. In this instance, a variety of DL-based models, including CNN, Bi-LSTM, models with CRF at inference layer are used to train the model. The architecture of the DL-based model for Odia NER system depicted in Figure 3. Figure 3 outlined the stages required in creating a DL model in order to make it simple.

1. The model takes an Odia sentence as input.
2. In order to incorporate information pertaining to the character sequences of Odia words, neural encoders such as CNN and Bi-LSTM models are employed as character-level embeddings.
3. Pre-trained FastText Odia word vector is used to initialize for word-level embeddings.
4. A fully connected NN is then fed the combined character and word embeddings.
5. The output of the previous layer gets inputted into the word sequence layer as input.
6. The output of the final hidden layer of the word sequence layer is utilized as input for the inference layer (CRF or softmax) in order to make predictions over possible tags associated with each input word.

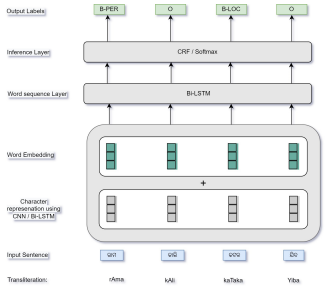

Figure 3: Architecture of Odia NER system using deep learning-based model

4 Experimental Results

4.1 Dataset Description

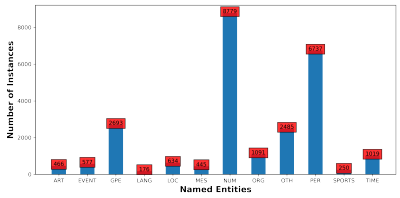

Figure 4: Class wise NEs in Odia NER dataset

For our experiment, we have used a subset of the public Odia NER corpus (Dalai et al., 2025). The Odia NER dataset includes the twelve NE types of PERSON, ORGANIZATION, GEOPOLITICAL ENTITY, LOCATION, EVENT, LANGUAGE, ART, SPORTS, NUMBER, TIME, MEASUREMENT, and OTHERS.

Sentence	Transliteration	POS Tags	NER Tags	Sentence	Transliteration	POS Tags	NER Tags
୨୦୦୨୧୬	2007re	NOUN	B-TIME	ଆଲିମ୍ପିକ୍	alimpik	PROPN	B-EVENT
ପାକିସ୍ତାନ	pAkistAna	PROPN	B-GPE	ପଦକ	padaka	NOUN	O
ବିପକ୍ଷରେ	bipakShare	NOUN	O	ବିଜେତା	bijetA	NOUN	O
୧୨	୧୨	NUM	B-NUM	ବିଜେନ୍ଦ୍ର	bijendara	PROPN	B-PER
ରନ୍	ran	NOUN	O	ସିଂହ	siMha	PROPN	I-PER
ଏଏରୁ	skora	NOUN	O	ରାଜ୍ୟଗୋଷ୍ଠୀ	rAjyagoShThI	NOUN	B-EVENT
କରିଥିବାବେଳେ	karithibAbeLe	VERB	O	କ୍ରୀଡ଼ା	krI.Da:_A	NOUN	I-EVENT
୨୦୦୩୧୬	2003re	NOUN	B-TIME	ବକ୍ସିଂର	baksiMra	NOUN	I-EVENT
ଅଟଲ୍ଟ୍ରିଲି	aShTreliA	PROPN	B-GPE	ଖୁବ୍ଟର	kwArTara	ADJ	O
ବିପକ୍ଷରେ	bipakShare	NOUN	O	ଫାଇଟରରେ	phAitARare	NOUN	O
ଶତକ	shataka	NOUN	B-NUM	ପ୍ରବେଶ	prabesha	NOUN	O
ହାସଲ	hAsala	NOUN	O	କରିଛନ୍ତି	kariChanti	VERB	O
କରିଥିଲେ	karithile	VERB	O	।	।	PUNCT	O
।	।	PUNCT	O				

Figure 5: A sample of Odia NER dataset

The dataset included 10,950 annotated sentences with a total of 158,947 tokens and 25,352 named entities. Figure 4 displays the statistics of the named entities in the Odia NER dataset. The dataset is split into three distinct parts: (1) the development set; (2) the training set that will be queried; and (3) the test set that will be evaluated. The distribution of the Odia NER corpus is shown in Table 1.

Table 1: **Odia NER corpus details**

Data	Number of Sentences	Number of tokens
Training	7660	1,11,250
Testing	1650	23,830
Development	1640	23,867
Total	10,950	1,58,947

4.2 Results

In order to verify the effectiveness and performance of our Deep-AL model, we have implemented different deep-learning techniques. Training the Odia NER models involved the usage of the Bi-LSTM classifier, which encoded words using a Bi-LSTM model and character level encoding using either of CNN or Bi-LSTM, and finally, the inference layer was handled by softmax or CRF tag decoder. We employed the conventional separation of the datasets, which included training, validation, and test data. Our Odia NER dataset was divided according to the usual 70% / 15% / 15% split, with 70% going to training and 15% each to validation and testing. The performance of the test dataset is used to determine parameters such as the number of iterations, learning rate, etc. To initialize the token, we employed a character embedding size of 30 and word embedding size of 300. Our complete deep learning system was trained with a stochastic gradient descent optimizer with a learning rate of 0.001, batch size of 128, and dropout rate of 30%. Our model comprised 300-dimensional word embeddings (WE) and utilized the pre-trained FastText model. We trained the models for 30 epochs. The number of active learning iterations was set at 25 due to the observation

that each algorithm does not exhibit significant improvement after 20 iterations.

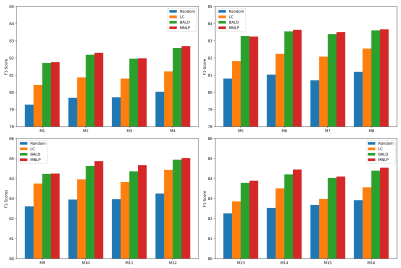

Figure 6: F1-Score of different models with active learning strategies, M1: Bi-LSTM + Softmax, M2: Bi-LSTM + Softmax + POS , M3: Bi-LSTM + CRF , M4: Bi-LSTM + CRF + POS , M5: WE + Bi-LSTM + Softmax , M6: WE + Bi-LSTM + Softmax +POS , M7: WE + Bi-LSTM + CRF , M8: WE + Bi-LSTM + CRF + POS , M9: WE + CharCNN + Bi-LSTM + Softmax , M10: WE + CharCNN + Bi-LSTM + Softmax + POS, M11: WE + CharCNN + Bi-LSTM + CRF, M12: WE + CharCNN + Bi-LSTM + CRF + POS , M13: WE + CharBi-LSTM + Bi-LSTM + Softmax , M14: WE + CharBi-LSTM + Bi-LSTM + Softmax + POS , M15: WE + CharBi-LSTM + Bi-LSTM + CRF , M16: WE + CharBi-LSTM + Bi-LSTM + CRF + POS.

The Deep-AL process begins with a random selection of samples from the training dataset, on which the model was trained. Following this, the learning process consists of numerous iterations. At the beginning of each round, the Deep-AL algorithm selects the unannotated sentences from the data pool to be annotated based on a specified budget. After the samples are labeled, they are incorporated into the training data, and the data pool and training set is then updated. Therefore, the model parameters are modified through training on the current training dataset before proceeding to the next iteration. We initiated our experiments with 2% of the training data from the Odia NER corpus that was labeled. In addition, the same number of data was added at each learning iteration, and the precision, recall, and F1 score are used to evaluate the model performance on the testing set. Furthermore, we detailed the performance of our model following its completion of training. Each experiment was repeated five times, and the average F-scores are recorded. The results depicted in Figure 7 demonstrate that all active learning algorithms outperform the random baseline in the Odia NER corpus. Additionally, the results indicate that the MNLP approach displays superior performance when compared to other active learning strategies on Odia NER dataset.

Table 2 depicted the results of our comparative analysis of the Odia NER performance of several models with MNLP active learning strategy. The graph depicted in Figure 7 displays F1 scores on the y-axis and the proportion of tagged words used for training on the x-axis. The results indicate that active learning methods utilizing only 38% of the training data on the Odia NER dataset were able to achieve 99% of the performance of the deep learning model that was trained with complete data. Table 3 presents the precision, recall, and F1 score for each distinct named entity class in our optimal Odia named entity recognition system.

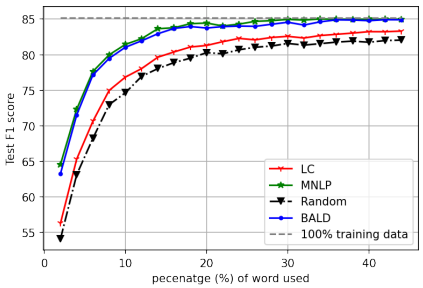

Figure 7: F1 score on the test dataset, in terms of the number of words labeled

Table 2: **Precision, Recall and F1 score of different models on Test data**

Model	Usage of POS information	Precision	recall	F1-Score
Bi-LSTM + Softmax	NO	82.67	80.87	81.76
Bi-LSTM + CRF		82.85	81.07	81.98
WE + Bi-LSTM + Softmax		84.23	82.29	83.25
WE + Bi-LSTM + CRF		84.56	82.48	83.51
WE + CharCNN + Bi-LSTM + Softmax		85.11	83.40	84.25
WE + CharCNN + Bi-LSTM + CRF		85.36	84.00	84.67
WE + CharBi-LSTM + Bi-LSTM + Softmax		85.03	82.78	83.89
WE + CharBi-LSTM + Bi-LSTM + CRF		85.18	83.05	84.10
Bi-LSTM + Softmax	YES	83.28	81.34	82.30
Bi-LSTM + CRF		83.92	81.49	82.69
WE + Bi-LSTM + Softmax		84.63	82.68	83.64
WE + Bi-LSTM + CRF		84.55	82.80	83.67
WE + CharCNN + Bi-LSTM + Softmax		85.29	84.45	84.87
WE + CharCNN + Bi-LSTM + CRF		85.76	84.29	**85.02**
WE + CharBi-LSTM + Bi-LSTM + Softmax		85.11	83.80	84.45
WE + CharBi-LSTM + Bi-LSTM + CRF		85.23	83.86	84.54

Table 3: **Label wise score of WE+CharCNN+Bi-LSTM+CRF model on Test data**

Name entity	Precision	Recall	F1-score
ART	87.16	82.92	84.99
EVENT	81.29	78.36	79.80
GPE	91.90	89.63	90.75
LANG	90.32	90.11	90.21
LOC	79.51	75.79	77.61
MES	92.32	94.56	93.43
NUM	91.78	93.97	92.47
ORG	82.18	76.39	79.18
OTH	82.37	77.46	79.84
PER	89.40	91.62	90.50
SPORTS	66.67	71.36	68.94
TIME	94.29	89.43	91.80
Macro average	85.76	84.29	85.02

5 Conclusion and Future Work

In this work, we presented a cost-effective and resource-efficient NER system for the low-resource Odia language using a Deep-AL framework. By integrating deep learning architectures with active sample selection strategies, we addressed the challenges posed by limited annotated data, high labeling costs, and the linguistic complexity of Odia. Our proposed approach demonstrated that high performance can be achieved up to an F1 score of 85.02% using only 38% of the annotated data required by traditional deep learning models, thereby reducing annotation costs by approximately 62%. Through extensive experimentation, we showed that incorporating character-level features, pretrained FastText embeddings, POS information, and a CRF-based inference layer led to improved model performance. The results indicate that our approach not only outperforms standard supervised methods but also demonstrates scalability and efficiency, making it suitable for similar low-resource language settings. Despite promising outcomes, there are several directions for future enhancement of this work: The framework can be adapted for other low-resource Indian languages, facilitating cross-lingual and multilingual NER systems with shared architectures and embeddings. Future experiments may involve incorporating transformer-based architectures such as BERT, XLM-R, or IndicBERT for richer contextual representation and better generalization. Although we evaluated common strategies like LC, BALD, and MNLP, future work could explore more advanced, or hybrid query strategies tailored specifically for NER in agglutinative and morphologically rich languages like Odia. This research represents a significant step toward democratizing NLP technology for low-resource languages and highlights the practical feasibility of deploying scalable, accurate NER systems under constrained resources.

References

Hamed H Aghdam, Abel Gonzalez-Garcia, Joost van de Weijer, and Antonio M López. 2019. Active learning for deep detection neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3672–3680.

Rayner Alfred, Leow Chin Leong, Chin Kim On, and Patricia Anthony. 2014. Malay named entity recognition based on rule-based approach. *International Journal of Machine Learning and Computing*, 4(3).

Vinayak Athavale, Shreenivas Bharadwaj, Monik Pamecha, Ameya Prabhu, and Manish Shrivastava. 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.

Sudhansu Bala Das, Atharv Biradar, Tapas Kumar Mishra, and Bidyut Kr. Patra. 2023. Improving multilingual neural machine translation system for indic languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–24.

Sudhansu Bala Das, Divyajyoti Panda, Tapas Kumar Mishra, Bidyut Kr. Patra, and Asif Ekbal. 2024. Multilingual neural machine translation for indic to indic languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(5):1–32.

Rakesh Balabantaray, Suprava Das, and Kshirabdhi Tanaya Mishra. 2013. Case study of named entity recognition in odia using crf++ tool. *International Journal of Advanced Computer Science and Applications*, 4(6).

Krisztian Balog, Pavel Serdyukov, and Arjen P de Vries. 2010. Overview of the trec 2010 entity track. Technical report, NORWEGIAN UNIV OF SCIENCE AND TECHNOLOGY TRONDHEIM.

Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Fifth Conference on Applied Natural Language Processing*, pages 194–201.

S Biswas, MK Mishra, S Acharya Sitanath_biswas, and S Mohanty. 2010. A two stage language independent named entity recognition for indian languages. *IJCSIT) International Journal of Computer Science and Information Technologies*, 1(4):285–289.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the association for computational linguistics*, 4:357–370.

Deepti Chopra, Nusrat Jahan, and Sudha Morwal. 2012. Hindi named entity recognition by aggregating rule based heuristics and hidden markov model. *International Journal of Information*, 2(6):43–52.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Tusarkanta Dalai, Anupam Das, Tapas Kumar Mishra, and Pankaj Kumar Sa. 2025. Odner: Ner resource creation and system development for low-resource odia language. *Natural Language Processing Journal*, 11:100139.

Tusarkanta Dalai, Tapas Kumar Mishra, and Pankaj K Sa. 2023. Part-of-speech tagging of odia language using statistical and deep learning-based approaches. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*

Tusarkanta Dalai, Tapas Kumar Mishra, and Pankaj K Sa. 2024. Deep learning-based pos tagger and chunker for odia language using pre-trained transformers. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(2):1–23.

Bishwa Das and Srikanta Patnaik. 2013. Name entity recognition for odia language using support vector machine.

Bishwa Ranjan Das, Srikanta Patnaik, Sarada Baboo, and Niladri Sekhar Dash. 2015. A system for recognition of named entities in odia text corpus using machine learning algorithm. In *Computational Intelligence in Data Mining-Volume 1*, pages 315–324. Springer.

Sudhansu Bala Das, Samujjal Choudhury, Tapas Kumar Mishra, and Bidyut Kr Patra. 2025a. Investigating the effect of backtranslation for indic languages. In *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 152–165.

Sudhansu Bala Das, Divyajyoti Panda, Tapas Kumar Mishra, and Bidyut Kr Patra. 2025b. Statistical machine translation for indic languages. *Natural Language Processing*, 31(2):328–345.

Gianluca Demartini, Tereza Iofciu, and Arjen P de Vries. 2009. Overview of the inex 2009 entity ranking track. In *International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 254–264. Springer.

Di Feng, Xiao Wei, Lars Rosenbaum, Atsuto Maki, and Klaus Dietmayer. 2019. Deep active learning for efficient training of a lidar 3d object detector. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 667–674. IEEE.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR.

Ralph Grishman and Beth M Sundheim. 1996. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Denis Gudovskiy, Alec Hodgkinson, Takuya Yamaguchi, and Sotaro Tsukizawa. 2020. Deep active learning for biased datasets via fisher kernel self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9041–9049.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2018. A deep neural network based approach for entity extraction in code-mixed indian social media text. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Vishal Gupta and Gurpreet Singh Lehal. 2011. Named entity recognition for punjabi language text summarization. *International journal of computer applications*, 33(3):28–32.

Xiao Lin and Devi Parikh. 2017. Active learning for visual question answering: An empirical study. *arXiv preprint arXiv:1711.01732*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

Andrew Mccallum. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proc. of CoNLL, 2003*.

Sudha Morwal, Nusrat Jahan, and Deepti Chopra. 2012. Named entity recognition using hidden markov model (hmm). *International Journal on Natural Language Computing (IJNLC) Vol*, 1.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Kashif Riaz. 2010. Rule-based named entity recognition in urdu. In *Proceedings of the 2010 named entities workshop*, pages 126–135.

Sujan Kumar Saha, Pabitra Mitra, and Sudeshna Sarkar. 2012. A comparative study on feature reduction approaches in hindi and bengali named entity recognition. *Knowledge-Based Systems*, 27:322–332.

Erik Tjong Kim Sang and Fien De Meulder. 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*, pages 142–147.

B Sasidhar, PM Yohan, A Vinaya Babu, and A Govardhan. 2011. Named entity recognition in telugu language using language dependent features and rule based approach. *International Journal of Computer Applications*, 22(8):30–34.

Christopher Schröder and Andreas Niekler. 2020. A survey of active learning for text classification using deep neural networks. *arXiv preprint arXiv:2008.07267*.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *proceedings of the 2008 conference on empirical methods in natural language processing*, pages 1070–1079.

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

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66.

Ye Zhang, Matthew Lease, and Byron Wallace. 2017. Active discriminative text representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Jingbo Zhu and Eduard Hovy. 2007. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 783–790.