IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 7, July 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

9940 572 462 | 6381 907 438 | ijircce@gmail.com | www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 7, July 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1007095 |

Comparative Study of Maithili Pre-Trained Embeddings on Named Entity Recognition Task

Shail Bala, Sujan Kumar Saha

Department of Computer Science and Engineering, Birla Institute of Technology, Mesra Ranchi, India

Department of Computer Science and Engineering, Birla Institute of Technology, Mesra Ranchi, India

ABSTRACT: This work aims to develop a Deep Learning based system for Named Entity Recognition (NER) in Maithili. Although NER is a popular task and NER systems have been developed in various Indian languages, much effort is not given to Maithili NER development. This study compares the performance of various pre-trained Maithili embeddings by evaluating them using the state-of-the-art model based on BiLSTM-CRF on the NER task. As Maithili is a low-resource language, we scraped the internet to generate a raw Maithili corpus for creating word embeddings. The scraped data was preprocessed, split into sentences, and deduplicated. About 4.36% of duplicate sentences were present in the corpus, which was removed. The final raw corpus contains about 52 lakh tokens and 30 thousand sentences. This corpus is used to learn Maithili embeddings. We compare Word2Vec and FastText embeddings and combine them with Flair character embeddings. We used an in-house NER dataset that contains about 2.5 lakh tokens for these experiments. Experimental results show that a combination of Word2Vec and character embedding perform the best among all combinations, outperforming FastText embeddings, which are subword embeddings.

KEYWORDS: Maithili NER, Pre-trained embedding, BiLSTM- CRF, Word2Vec, FastText

I. INTRODUCTION

Natural language processing (NLP) is a subfield of computer science and artificial intelligence concerned with the processing and analysis of large amounts of natural language data. The goal of NLP is to understand natural languages, including their contextual nuances. The task of Named Entity Recognition (NER) is to identify and categorize named entities (NE) present in a text into predefined categories like names of people, location, organization, dates etc. For understanding of a language, identifying and differentiating between the category of names becomes pivotal. So, NER systems have applications in areas like information extraction, question answering, and machine translation.

Maithili is an Indo-Aryan language native to parts of India and Nepal. In India, it is spoken in Bihar and northeastern Jharkhand, and is one of the 22 scheduled official languages of India. In Nepal, it is native to eastern Terai region and is the second most spoken language of the country. The language is predominantly written in Devanagari, but has its own script called Mithilakshar or Tirhuta. It ranks 40th among the most spoken languages of the world, while it occupies the 16th position in the list of the most spoken languages in India.

Embedding is a language modeling technique used for mapping words or sequences of characters to vectors of real numbers. Word embeddings encode the words into numeric forms while maintaining the word relationships in that space, like meaning, morphology, context, or some other kind of relationship. The study aims to compare various pre-trained embeddings on Maithili.

In this work we present a deep-learning based model for Maithili NER. We have also performed a comparative study on the performance of various word and character embeddings on Maithili NER task. We have prepared the baseline model using BiLSTM-CRF and an in-house Maithili NER corpus containing around 250K words. Then we searched for pre-trained Maithili embeddings. We found only one embedding that supported Maithili. This was one FastText embedding trained on 157 languages from Wikipedia dumps and Common Crawl data. To generate few other embeddings, we created our own raw Maithili corpus scraped from various web sources, as no raw corpus was open online. We trained Word2Vec, FastText and Flair character embeddings on this raw corpus of various dimensions. Then we ran various experiments with these embeddings on the Maithili NER task. Finally, we achieved an F1-score of 89.64 with a combination of Word2Vec and Flair character embeddings. Word2Vec with dimension 300 gave the best F1 score with Flair character embedding of dimension 30. Corresponding precision and recall values were 93.16 and 86.38 respectively.

The details of the experimental setup, data, embeddings, results and discussions are presented in the subsequent

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 7, July 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1007095 |

sections of the paper.

II. RELATED WORK

The task of Named Entity Recognition was coined in 1995 in Message Understanding Conference-6 (MUC-6) for the English language [1]. Since then various NER tasks and systems have been proposed. Early NER systems employed handcrafted rules, lexicons, orthographic features and gazetteers. But these systems are very time-consuming and require linguistic experts to form rules. This was followed by statistical NER systems based on feature-engineering and later, machine learning based systems were introduced. The first Neural Network based NER system was proposed in 2011 for English [2], with minimal feature engineering. Recent trends show the use of deep learning models with [17] establishing the BiDirectional Long Short Term Memory (BiLSTM) with CRF layer as State-of-the-art NER system for English producing near-human performance. The best systementering MUC-7 scored 92% recall and 93% precision [3].

The first NER system in Hindi was introduced in 2003 [4] using Conditional Random Fields (CRF) and feature induction. Later, many systems were proposed ranging from rule based system [5], to Hidden Markov Model [6] and Maximum Entropy model [7]. In another paper, the authors compared the performance of CRF and Support Vector Machine (SVM) [8]. Starting 2016, deep learning models for

Hindi NER were proposed which gave better performance with no explicit encoding of linguistic features or rules. The BiLSTM model is found to perform well [9,10] along with deep contextualized embeddings like BERT and ELMO [11]. Indian languages are morphologically rich and unlike the English language, where capitalized letters give a hint of named entities, there is no support as such. Also, most Indianlanguages are free-order and provide added difficulty when identifying NE.

Maithili is considered a low-resource language but efforts have been made recently to develop Maithili corpora. The first effort towards a Maithili NER system was done in 2020 with a corpus size of 157468 annotated using 22 entity labels, though the data is not open. The reported F1-score was 73.19 using a baseline CRF model [12]. Another corpus with 5 Named Entities gave the best F1 score of 87.78 with Bi- LSTM-CRF model employing character embeddings on a 200024 sized dataset [13]. The performance improved further with the employment of gazetteers to an F1-score of 91.6. In Maithili language, names also get inflected and there are many ambiguities as common nouns are also used as names. This makes Maithili NER task more challenging than in English or Hindi.

We observed that BiLSTM with CRF output layer performs the best for Maithili and therefore, we have taken this model for our comparative study.

Word Embeddings have been reported to improve performance over sequence labeling tasks like Part-of-speech (POS) tagging and NER tasks [24]. A survey on Word embeddings [25] points out that embeddings act as extra features for downstream NLP tasks. The IndicNLP suite [23] released the first massive NLP corpora for 11 indian languages, but the support for Maithili is not there in it.

III. EXPERIMENTAL SETUP AND DATASET

This section presents our work on development of a Maithili NER system. Figure 1 presents a summarized workflow of the process we follow for the development. The details of various embeddings are discussed in Section IV and the details of the NER corpus and raw corpus is present below.

Now, we present the datasets used and our effort at datasetcreation along with the raw corpus used for training Maithili embeddings.

A. NER Corpus

In this study we have primarily used the NER corpus developed in [13]. It contains NE with 5 classes: Person, Location, Organization, Number, and Miscellaneous. The corpus contains around 2 lakh annotated tokens. Person, Location, and Organization are the common categories in NER tasks. The Number class contains numericals, units, measurements, monetary amounts, and date. The Miscellaneous class includes names of Festivals, Events, Books, Movies, Languages, and Religion/caste etc. We converted the tags used in the dataset to the BIO format [22] for each class tag. Table I lists the conversion used for each class.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 7, July 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1007095 |

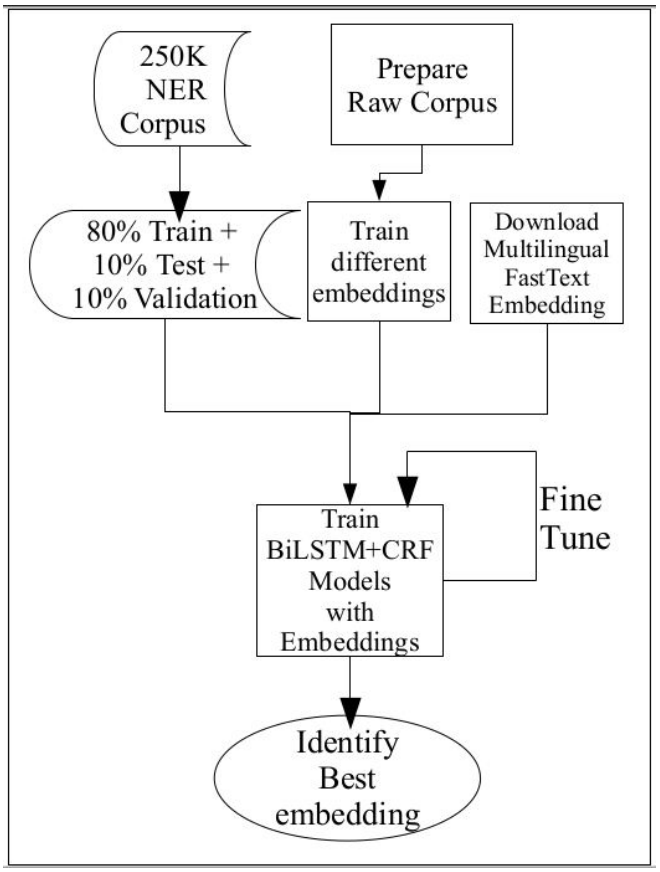

Fig. 1. Flow Diagram of the System

TABLE I. NER TAGS USED

SNo	Named Entities		
	Named Entity Class	*Original Tag*	*BIO Tag given*
1	Person-Begin	#PB	B-PER
2	Person-Interior	#PI	I-PER
3	Location-Begin	#LB	B-LOC
4	Location-Interior	#LI	I-LOC
5	Organization-Begin	#OB	B-ORG
6	Organization-Interior	#OI	I-ORG
7	Number-Begin	#NB	B-NUM
8	Number-Interior	#NI	I-NUM
9	Miscellaneous-Begin	#MB	B-MISC
10	Miscellaneous-Interior	#MI	I-MISC
11	Other	#O	O

B. NER Corpus Extension

As the authors of [13] mentioned that the corpus is not sufficient, we started our work by extending the available

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 7, July 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1007095 |

corpus. We have manually annotated an additional corpus containing around 56 thousand tokens with BIO format consistent with the former dataset. So, the final NER dataset contains around 250K tokens tagged with five NER classes.

This corpus was splitted into Train, Validation and Test sets for the experiments. The sentences in the test set were chosen randomly from the whole corpus. The details of the Train, Validation and Test sets are given in Table II.

TABLE II. TRAIN-TEST-VALIDATION SPLIT OF COMBINED NER DATASET

Split	*Tokens*	*Sentences*
Train	208793	10063
Test	21309	964
Validation	20059	855
Total	250161	11882

C. Raw Corpus for learning pre-trained embeddings

We could not find any raw corpus online to train Maithili embeddings. So we scraped various Maithili news websites, Wikipedia and literature to generate our raw corpus. We wrote our scraper using Selenium, BeautifulSoup and Pandas libraries in Python. For some websites, we also used Boilerpyto extract text [21]. The web scraper scraped about 24,000 pages from 18 websites. This data was preprocessed and deduplicated to produce the raw corpus.

As part of preprocessing, the following aspects were treated and/or removed from corpus:

- Emojis in text were removed
- Emails/URLs present were replaced with a tag like <Email> or <URL>
- HTML/XML tags were removed
- Multiple spaces and newlines were truncated
- A newline beginning with a symbol other than ‘ or “, was removed.
- Combinations like ।. or ।, were replaced with ।followed by a space. Similarly for Exclamation and Question marks.
- English sentences were removed.
- Multiple copies of one symbol were replaced with a single copy of that symbol (,, -> ,)

Then the raw text was split based on । (Full Stop in Devanagari Script), ! (Exclamation Mark) or ? (Question Mark). Once again, multiple spaces and newlines were treated.

After preprocessing, the data was in the format of one line containing exactly one processed sentence. Before training word embeddings or models, it is required to remove duplicate entries for improved performance. So, the corpus was then deduplicated to ensure only unique sentences remain. The duplicates were 4.36% in total. After deduplication, total words in the raw corpus were 52,74,580 and total unique words were 3,25,067.

IV. PRETRAINED EMBEDDINGS

Our study aims to compare various pre-trained Maithili embeddings on the model BiLSTM-CRF for the NER task. We used one FastText embedding [15] available online for Maithili. No other embeddings were available that included Maithili. So, we trained our own Word2Vec, FastText and Flair character embeddings on the raw corpus with various dimension sizes.

A. Word2Vec Embeddings

Word2Vec is a model that embeds words in a lower- dimensional vector space using a shallow neural network. Word2Vec uses the idea that words which occur in similar contexts are similar. Thus, they can be clustered together. There are two models introduced: Continuous Bag of Words (CBOW) and Skipgram [17, 20]. The result is a set of word-

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 7, July 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1007095 |

vectors where vectors close together in vector space have similar meanings based on context, and word-vectors distant to each other have differing meanings.

Four Word2Vec embeddings were trained on the Maithili corpus with dimension size 100, 200, 300 and 400. The window size and the minimum frequency value of words to be present in the vocabulary were kept as 5. For this purpose,Python's Gensim library was used. The IndicNLP [20] librarywas used to tokenize the sentences before giving them to Gensim's embedding generator.

B. FastText Embeddings

FastText is another word level embedding like Word2Vecbut it is based on subword level information for representing words as vectors. In FastText, the SkipGram architecture from Word2Vec was operated at a character n-gram level, using a bag of character n-grams. Here, words are representedby a combination of the character n-gram vectors.
We have used two type of FastText embeddings, which are given below:

a) Custom Trained

We used the FastText [14] library to train our own FastText embeddings. We trained two embeddings with dimensions 100 and 300, using CBOW and subwords rangingin between 3-6, with character n-grams of length 5.

b) FastText Multilingual Word Vectors

We downloaded the FastText vectors [15] trained over Wikipedia and Common Crawl data of 157 languages including Maithili. These models were trained using CBOW with position-weights, in dimension 300, with character n- grams of length 5, a window of size 5 and 10 negatives.

C. Flair Character Embeddings

Flair embeddings [16] are deep contextual string embeddings. In this embedding each of the letters in the words are sent to the Character Language Model and then the input representation is taken out from the forward and backward LSTMs. The representation for each word is generated based on the words present before and after it. It thus gives different embeddings for the same word dependingon its surrounding text.

We varied the character embedding dimensions from 10 to 40 and found the embeddings with dimension 30 often produced the best result. In the result section we have reported the values with two character embeddings dimensions: 10 and30.

V. RESULTS AND DISCUSSION

This section presents the results we obtained in our experiments with the Maithili NER task.

The deep learning model used for performing this comparison uses the BiLSTM-CRF model. We used the libraries FlairNLP [18] and Kashgari [19] for implementing the models with various embeddings. The embeddings are used as the input layer to this system. We combined the two NER datasets and then split them into train, test and validation sets of 80%, 10% and 10% samples, randomly. With different combinations of embeddings, we tuned the system and thefinal F1-Score was measured on the test set.

The best F1-score of 89.64 was achieved with a combination of Word2Vec and Flair character embeddings. Corresponding precision and recall values are 93.16 and
86.38 respectively. Word2Vec with dimension 300 gave the best F1 score with Flair character embedding of dimension
30. In general, we observed, Word2Vec embeddings performed better than FastText embeddings. Again, we observed from the results that custom FastText embedding performed better than the Multilingual vectors downloaded.

Table II presents the F1-score along with precision and recall of all combinations of pre-trained embeddings used for this study. Most of the models achieved a higher precision and a lower recall. The lower recall value indicates that the system will perform better if more training resources can be provided.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 7, July 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1007095 |

TABLE III. PERFORMANCE OF VARIOUS EMBEDDINGS

SNo	*Embedding Used*	*Dimension*	*Precision*	*Recall*	*F1 Score*
1	FastText Multilingual	300	78.91	66.3	72.06
2	Custom FastText	100	82.06	68.2	74.28
3	Custom FastText	300	80.06	69.57	74.34
4	Word2Vec	100	89.62	79.54	84.28
5	Word2Vec	200	91.36	80.56	85.62
6	Word2Vec	300	91.08	82.1	86.35
7	Word2Vec	400	90.02	79.9	84.66
8	FastText + FlairChar	300 + 10	84.8	68.7	75.91
9	FastText + FlairChar	300 + 30	83.56	70.51	76.44
10	Custom FastText + Flair Char	100 + 10	87.8	70.96	78.71
11	Custom FastText + FlairChar	100 + 30	86.8	72.48	78.9
12	Custom FastText + FlairChar	300 + 10	87.42	72.6	79.32
13	Custom FastText + FlairChar	300 + 30	89.23	75.1	81.52
14	Word2Vec +Flair Char	300 + 10	90.82	85.17	87.90
15	Word2Vec +Flair Char	300 + 30	93.16	86.38	89.64

Class wise performances of FastText, Word2Vec,

TABLE V. CLASS WISE F1-SCORE OF WORD2VEC EMBEDDINGS

Class	*Dim 100*	*Dim 200*	*Dim 300*	*Dim 400*
Person	82.5	78.73	83.57	83.75
Location	81.22	78.75	80.58	80.13
Number	90.3	91.57	92.56	89.72
Organization	84.83	85.45	84.6	84.26
Miscellaneous	85.58	93.62	91.46	86.37

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 7, July 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1007095 |

TABLE VI. CLASS WISE F1-SCORE OF CUSTOM FASTTEXT +FLAIR EMBEDDINGS

Class	*Dim 100 + 10*	*Dim 100 + 30*	*Dim 300 + 10*	*Dim 300 + 30*
Person	87.3	80.5	87.7	85.46
Location	62.4	73.62	69.75	72.97
Number	72.7	75.82	78.32	88.5
Organization	68.45	77.45	77.89	79.5
Miscellaneous	85.06	88.2	80.9	81.2

TABLE VII. CLASS WISE F1-SCORE OF WORD2VEC + FLAIREMBEDDINGS

Class	*Dim 300 + 10*	*Dim 300 + 30*
Person	85.45	85.73
Location	80.98	83.75
Number	91.34	94.57
Organization	88.61	87.45
Miscellaneous	89.43	95.7

We also like to compare our results with the Maithili NERresults presented in [13]. They achieved the highest accuracyof 91.60 F1-score in their Maithili NER system. However, inorder to achieve this value, they used a few gazetteer lists. Without the gazetteer lists, the value is an F1-score of 87.78.In our system, we achieved an F1-score of 89.64. So, if we solely rely on embeddings, our system performed better thanthe system by Priyadarshi and Saha [13]. This is because of the custom character and word embeddings we trained using our collected raw corpora. These values prove that using the appropriate word and character embeddings can help improveperformance.

combination of FastText and Flair, and combination of Word2Vec and Flair are given in Table III, IV, V and VI respectively.

TABLE IV. CLASS WISE F1-SCORE OF FASTTEXT EMBEDDINGS

Class	*FastText Multilingual*	*Custom FastText 100*	*Custom FastText 300*
Person	68.84	69.71	70.25
Location	65.03	65.98	66.5
Number	82.4	74.68	76.2
Organization	64.22	65.63	75.2
Miscellaneous	79.8	85.02	85.2

VI. CONCLUSION

In this paper we proposed a new Maithili NER system. The system uses Bi-LSTM-CRF as the baseline model, and then uses various word and character embeddings to improvethe performance. We also present a comparative study of a few word embedding and character embedding techniques onMaithili NER. The best F1-score of 89.64 was achieved witha combination of Word2Vec embedding of dimension 300 and Flair character embedding of dimension 30.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 7, July 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1007095 |

We observed that Word2Vec performed better than FastText embeddings. And the use of contextualized character embeddings combined with word embeddings gave best performance. Custom word embedding trained using own corpus collection may provide better performance than pre-trained existing vectors. Maithili is a highly inflectional language where named entities are also inflected. The character embeddings capture these effectively and thus increase the performance of the system.

REFERENCES

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

[2] Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. "Natural language processing (almost) from scratch." Journal of machine learning research 12, no. ARTICLE (2011): 2493-2537.

[3] Black, William J., Fabio Rinaldi, and David Mowatt. "FACILE: Description of the NE System Used for MUC-7." In Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998. 1998.

[4] Li, Wei, and Andrew McCallum. "Rapid development of Hindi named entity recognition using conditional random fields and feature induction." ACM Transactions on Asian Language Information Processing (TALIP) 2, no. 3 (2003): 290-294.

[5] Kaur, Yavrajdeep, and Er Rishamjot Kaur. "Named Entity Recognition (NER) system for Hindi language using combination of rule based approach and list look up approach." Int. J. Sci. Res. Manag.(IJSRM) 3, no. 3 (2015): 2300-2306.

[6] Saha, Sujan Kumar, Sanjay Chatterji, Sandipan Dandapat, Sudeshna Sarkar, and Pabitra Mitra. "A hybrid named entity recognition system for south and south east asian languages." In Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages. 2008.

[7] Chopra, Deepti, Nisheeth Joshi, and Iti Mathur. "Named entity recognition in Hindi using hidden Markov model." In 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT), pp. 581-586. IEEE, 2016.

[8] Krishnarao, Awaghad Ashish, Himanshu Gahlot, Amit Srinet, and D.
S. Kushwaha. "A comparative study of named entity recognition for Hindi Using sequential learning algorithms." In 2009 IEEE International Advance Computing Conference, pp. 1164-1169. IEEE, 2009.

[9] Shah, Bansi, and Sunil Kumar Kopparapu. "A Deep Learning approach for Hindi Named Entity Recognition." arXiv preprint arXiv:1911.01421 (2019).

[10] Singh, Sarika, Shashank Patel, Yash Shah, Rucha Nargunde, and Jyoti Ramteke. "Context-Based Deep Learning Approach for Named Entity Recognition in Hindi." In 2021 International Conference on Communication information and Computing Technology (ICCICT), pp. 1-5. IEEE, 2021.

[11] Litake, Onkar, Maithili Sabane, Parth Patil, Aparna Ranade, and Raviraj Joshi. "Mono vs Multilingual BERT: A Case Study in Hindi and Marathi Named Entity Recognition." arXiv preprint arXiv:2203.12907 (2022).

[12] Mundotiya, Rajesh Kumar, Shantanu Kumar, Umesh Chandra Chaudhary, Supriya Chauhan, Swasti Mishra, Praveen Gatla, and Anil Kumar Singh. "Development of a Dataset and a Deep Learning Baseline Named Entity Recognizer for Three Low Resource Languages: Bhojpuri, Maithili and Magahi." arXiv preprint arXiv:2009.06451 (2020).

[13] Priyadarshi, Ankur, and Sujan Kumar Saha. "The first named entity recognizer in Maithili: Resource creation and system development." Journal of Intelligent & Fuzzy Systems, Vol. 41 (2021) 1083–1095.

[14] Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. "Enriching word vectors with subword information." Transactions of the association for computational linguistics 5 (2017): 135-146.

[15] Mikolov, Tomas, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. "Advances in pre-training distributed word representations." arXiv preprint arXiv:1712.09405 (2017).

[16] Akbik, Alan, Duncan Blythe, and Roland Vollgraf. "Contextual string embeddings for sequence labeling." In Proceedings of the 27th international conference on computational linguistics, pp. 1638-1649. 2018.

[17] Huang, Zhiheng, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging." arXiv preprint arXiv:1508.01991 (2015).

[18] Akbik, Alan, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. "FLAIR: An easy-to-use framework for state-of-the-art NLP." In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pp. 54-59. 2019.

[19] Eliyar Eziz. “Kashgari.” In GitHub repository https://github.com/BrikerMan/Kashgari. 2019

[20] Kunchukuttan, Anoop, Divyanshu Kakwani, Satish Golla, Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. "Ai4bharat- indicnlp corpus: Monolingual corpora and word embeddings for indic languages." arXiv preprint

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 7, July 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1007095 |

arXiv:2005.00085 (2020).

[21] Kohlschütter, Christian, Peter Fankhauser, and Wolfgang Nejdl. "Boilerplate detection using shallow text features." In Proceedings ofthe third ACM international conference on Web search and data mining, pp. 441-450. 2010.

[22] Sang, Erik F., and Jorn Veenstra. "Representing text chunks." arXiv preprint cs/9907006 (1999).

[23] Kakwani, Divyanshu, Anoop Kunchukuttan, Satish Golla, N. C. Gokul, Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. "IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages." In Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4948-4961. 2020.

[24] Qu, Lizhen, Gabriela Ferraro, Liyuan Zhou, Weiwei Hou, Nathan Schneider, and Timothy Baldwin. "Big data small data, in domain out- of domain, known word unknown word: The impact of word representation on sequence labelling tasks." arXiv preprint arXiv:1504.05319 (2015).

[25] Almeida, Felipe, and Geraldo Xexéo. "Word embeddings: A survey." arXiv preprint arXiv:1901.09069 (2019).

[26] S.Kannadhasan and R.Nagarajan, Development of an H-Shaped Antenna with FR4 for 1-10GHz Wireless Communications, Textile Research Journal, DOI: 10.1177/00405175211003167, March 21, 2021, Volume 91, Issue 15-16, August 2021

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.165

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details