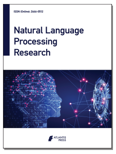
Natural Language Processing Research
Vol. 1(1-2), 2020, *pp.* 14–22
DOI: https://doi.org/10.2991/nlpr.d.201005.001; eISSN: 2666-0512
https://www.atlantis-press.com/journals/nlpr/



Research Article

A Framework for Named Entity Recognition for Malayalam—A Comparison of Different Deep Learning Architectures

R. Rajimol[1], V. S. Anoop[2,*]

[1] *Indian Institute of Information Technology and Management—Kerala (IIITM-K), Thiruvananthapuram, 695581, Kerala, India*
[2] *Rajagiri College of Social Sciences, Kochi, 683104, Kerala, India*

ARTICLE INFO

Article History

Received 03 June 2020
Accepted 22 Sep 2020

Keywords

Named entity recognition
Malayalam language
Deep learning
Information extraction
Low resource language computing
Indic language processing

ABSTRACT

Information extraction (IE) is the process of extracting relevant and useful patterns or information from unstructured data. Named entity recognition (NER) is a subtask of IE that identifies entities from unstructured text documents and organize them into different predefined categories such as person, location, organization, number, date, etc. NER is considered to be one of the important steps in natural language processing which may find direct applications in areas such as question answering (QA), entity linking, and co-reference resolution, to name a few. NER systems perform comparatively well in high-resource languages such as English but there is a lack of well-developed NER systems for low-resource languages such as Malayalam, which is an Indic language spoken in the state of Kerala, India. This work is an approach in this direction which makes use of deep learning (DL) techniques for developing a NER system for Malayalam. We have compared different DL approaches such as recurrent neural networks, gated recurrent unit, long short-term memory, and bi-directional long short-term memory and found that DL based approaches significantly outperform traditional shallow-learning based -approaches for NER. When compared with some state-of-the-art approaches our proposed framework is found to be outperforming in terms of precision, recall, and F-measure and could achieve an improved precision of 7.89% and 8.92% of F-measure.

1. INTRODUCTION

Today, we live in an era of information explosion where massive amounts of data in various forms such as text, audio, images, video, etc., are getting added to the Internet every minute. Analyzing and unearthing useful patterns from such humongous data is always a difficult task because of the unstructured format. A major share of such data is the unstructured text that does not have any specific format for representation. Information extraction (IE) is an area in artificial intelligence (AI) that deals with unearthing latent information primarily from unstructured text data. IE primarily extracts entities, key-phrases, topics, relations, and other patterns of interest that are mentioned in the unstructured text corpus. Named entity recognition (NER) is a subtask of IE that attempts to identify and classify named entities such as persons, organizations, locations, dates, etc., into predefined categories. Since the NER task was introduced as an information retrieval task firstly in the Message Understanding Conference (MUC) conducted in the year 1990, the research on NER has become the center of attraction. This caused many approaches getting reported in the information retrieval and natural language processing literature with varying degrees of success.

The approaches already reported for recognizing named entities can be primarily categorized into three types—(1) approaches based on language rules, (2) approaches based on machine learning, and (3) approaches that combine rules and machine learning or called hybrid approaches. The rule-based approaches use handcrafted custom rules created by linguistics experts which are highly specific to a particular language. The rules thus defined will identify the entities in the unstructured text [1]. One of the earliest approaches for NER was introduced in the year 1984 [2] where statistical methods were adopted for recognizing the entities based on the information content. A linguistics rule-based approach that attempts to identify common entities such as the names of persons, locations, and organizations was introduced in 1996 [3]. The main drawback of a rule-based approach to NER is that it requires in-depth knowledge about the grammar of a language [4]. The rule-based approaches were widely used in developing NER systems for many resource-rich languages in the era where machine learning and sophisticated computational capabilities were not available.

The advancements in software and hardware infrastructures along with the introduction of machine learning algorithms caused many approaches to getting introduced for recognizing named entities. The machine learning algorithms can be generally categorized as supervised and unsupervised. The supervised learning algorithms require a large amount of labeled or annotated training data and on the other side, there are unsupervised training algorithms that do not need the labeled data to work with. When analyzing the state-of-the-art, we may come to know that the vast majority of the reported approaches use supervised learning algorithms. Even though the unsupervised algorithms learn the patterns or

[*] *Corresponding author. Email: anoop@rajagiri.edu*

abstractions automatically, the level of accuracy will be less compared to the supervised models. The approaches reported in the NER literature that uses machine learning techniques are based on popular training algorithms such as Naive Bayes [5–8], support vector machine (SVM) [9–13], hidden Markov models (HMMs) [14–17], and artificial neural networks (ANNs) [18–21].

Very recently, deep learning (DL) has become the buzzword in machine learning and widely adopted in many application areas including natural language processing and information retrieval. DL architectures combine multiple layers which in turn can learn representations of data without handcrafting the features. No wonder, the DLrning architectures are now being widely used in building NER systems with its sophisticated architectures capable of representing text in character-level and word-level. While shallow-learning-based approaches use handcrafted features, DL approaches have already proved that it can better generate features thus improves learning for many areas where traditional machine learning approaches have degraded performances. The major methods used for training deep neural networks include recurrent neural networks (RNNs) [22,23], gated recurrent unit (GRU) [24,25], long short-term memory (LSTM) [26], bi-directional long short-term memory (BLSTM) [27], etc.

The natural language processing on low-resource languages is always considered to be a complex task as the research community is very small and the tools and available resources are very limited. Many of the Indic languages come into this category so as the case with the Malayalam language. Malayalam belongs to the Dravidian family of languages and mainly spoken in the State of Kerala in India. It is one of 22 scheduled languages in India with over 38 million native speakers. In 2013, Malayalam has designated a classical language status in India. The term Malayalam means "mountain region" and originates from the words "mala" (mountain) and "Alam" (place/region). It has a high influence of Sanskrit (an Indo-Aryan language) from this language Malayalam brings out its variety of words and compound alphabets. Malayalam belongs to the Dravidian family of languages along with Tamil, Kannada, and Telugu. Malayalam as a language being spoken in Kerala, the highest literacy state in India it can be proud of rich literature with a lot of literary works and is also a medium of many journals and newspapers with the highest circulation rate in India.

While named entity extraction from open domain unstructured text is a challenging task, working with the Malayalam is even harder. The Malayalam language is morphologically rich and highly agglutinative which makes it difficult to use for computational purposes. The features we use to recognize named entities in other languages (e.g., the capitalization feature to identify the name of a person, location, or organization, in the English language) cannot be used in Indic languages such as Malayalam as there is no capitalization used. As mentioned earlier, in a morphologically rich language such as Malayalam, the words are formed using multiple stems and affixes. This also makes the entity extraction a difficult task. Most importantly, the lack of resources such as tagged dataset, dictionaries, morphological tools, etc., make it challenging to work with IE tasks.

Even with all these challenges, there are some early attempts reported in the language computing literature that tries to develop NER approaches for the Malayalam language. The majority of those approaches used simple heuristics-based and statistical approaches for extracting named entities. Those approaches used a very limited set of features that operate on a very small training data set. Some of the recent approaches in this direction used ANNs that shows better performance compared to the earlier heuristics-based approaches. Recently, it is proved by machine learning research communities that DL is a better mechanism to perform computational tasks. Now, a vast majority of shallow-learning-based approaches have been converted into DL-based approaches and found to be outperforming the former. NER tasks for many languages have been now developed using DL techniques but the same has not been explored much in the case of Indic languages. So there exists a huge knowledge gap to bring in DL approaches for IE in Malayalam, specifically on the identification of named entities from the unstructured Malayalam corpus. This research is an attempt in this direction.

This paper proposes a DL-based approach for NER for the Malayalam language. We use different DL architectures such as GRU, RNN, LSTM, BLSTM, etc. to improve the NER system performance. The major contributions of this paper are summarized as follows:

- Discusses major issues with NER for low-resource languages such as Malayalam and lists out some of the recent but prominent state-of-the-art approaches in Malayalam NER.
- Proposes a DL-based approach for NER in Malayalam and compares different DL architectures for NER.
- Discusses the experimental setup and detailed analysis of the result with state-of-the-art approaches.
- The experimental results show that the BLSTM outperformed the other approaches and showed better precision, recall and F-measure when compared with the other DL-based methods. For the baseline comparison, outperformed the others with an improved precision of 7.89% and 8.92% of F-measure.

The remaining sections of this paper are organized as follows: Section 2 discusses some of the recent approaches reported in the area of NER for English and other resource-heavy languages, and also in Malayalam, which is a low-resource language. Section 3 introduces a DL-based framework for NER for Malayalam, and in Section 4 the experimental setup used is described. The conclusions and future works are discussed in Section 5.

2. RELATED WORK

This section details some of the recent and prominent approaches for NER. This section is divided into two subsections—one detailing some of the recent approaches reported in the NLP literature for English and other languages and the second subsection detailing the NER approaches reported in the literature for Malayalam NER.

2.1. NER in English and Other Resource Heavy Languages

There is a large number of approaches available for NER in English and other resource-heavy languages such as Chinese. Some of the very prominent works in these areas are discussed here. One of the earlier approaches that got significant attention was the use of the hidden Markov model (HMM) for recognizing named entities [28]. The authors have used HMM and an HMM-based chunk tagger for performing NE classification for names, times, and numerical quantities [28]. Another approach that used HMM for NER was reported [29] which is language independent. The authors claimed that their approach is generic and can be applied for NER in any language. In their proposed approach the states of the HMM model were not fixed and dynamic in nature [29]. A Maximum Entropy(MaxEnt)-based approach was reported in the literature that was proposed by Chieu and Ng [30]. The major difference of this approach with the earlier approaches was that it makes use of the information from the whole document to classify whether the word is a named entity or not. Their proposed approach was capable of capturing the global information directly and that leads to better classification accuracy [30].

Decision tree-based approaches for NER were also reported in the literature. One notable work that used decision tree induction was reported by Georgios Paliouras *et al.* [31]. In this work, the authors proposed a decision tree-based approach for classifying grammars where the construction of such a grammar is difficult and time-consuming. Using the MUC dataset, the authors could successfully validate their proposed approach using the C4.5 method [31]. A multilingual NER system that uses boosting and C4.5 was introduced by Szarvas *et al.* [32]. The authors used AdaBoostM1 and C4.5 to perform NER for Hungarian and English languages. On evaluating against the CoNLL dataset, their approach showed significant performance increase [32].

A recent approach was reported in the NLP literature that uses pooled contextualized embeddings for NER [46]. Proposed by Alan Akbik *et al.*, the proposed method dynamically aggregate contextualized embeddings of each unique string they encounter in the text. Then the authors used a pooling operation to distill a global word representation from all contextualized instances. An approach for NER on Arabic-English code-mixed data was proposed by Sabty *et al.* [47]. The authors claimed that they have the first annotated CM Arabic-English corpus for NER. Furthermore, they have constructed a baseline NER system using deep neural networks and word embedding for Arabic-English CM text and enhanced it using a pooling technique [47]. Another method that uses multilingual meta-embeddings for code-switching NER was proposed by Genta Indra Winata *et al.* [48]. In this work, the authors proposed multilingual meta-embeddings which is an effective method to learn multilingual representations by using monolingual pretrained embeddings [48]. This paper claims that their proposed method achieves state-of-the-art performance in a multilingual setting and has the generalization ability. An approach for cross-lingual transfer learning for Japanese language NER was reported in 2019 which was developed by Johnson *et al.* [49]. The authors focused on bootstrapping Japanese from English and then adopted a deep neural network and the best combination of weights to transfer is extensively investigated. Experiments are conducted on external datasets, as well as internal large-scale real-world ones to show that the proposed method achieves better NER accuracy [49].

2.2. Recent Approaches in Malayalam-NER

Very recent work in Malayalam NER was reported by Sreeja and Pillai [50] in which the authors performed an analysis of the challenges in NER and proposed a NER system for Malayalam using LSTM. Another recent work is reported by Ajees and Idicula in 2018 [33]. They used a CRF based approach, a probabilistic graphical model for sequence labeling. The system makes use of different features such as words, preceding words, the following words, suffixes of words, etc. The training is conducted on a corpus of 20615 sentences. In [33], Ajees and Idicula developed a NER system for Malayalam using neural networks in the year 2018. This system used a corpus of 20615 sentences. They used neural networks and word embedding approaches (Word2Vec). A Skip-gram based word embedding for NER was reported by Remmiya Devi *et al.* in 2016 [35]. A limited unlabelled dataset is used for this experiment. They extracted the named entities from Malayalam Social media. That was a closed domain. Shruthi in 2016 [40], reported another work in NER. They compared the performance of TnT and MEMM on NER. Limited sentences were used for training. Nita Patil *et al.* in 2016 [42] presented a survey of NER systems with respect to Indian and foreign languages. In this survey, they covered the study and observations related to approaches, techniques, and features to implement NER for various languages. A CRF-based approach was reported by Gowri Prasad *et al.* in 2015 [37] for Malayalam NER. The size of the data set for the training was too small. The supervised machine learning approaches highly depends on the size of the dataset used for training. So the limited size of the corpus was a drawback. This work was also in a closed domain (tourism). In [43], Sanjay *et al.* proposed a CRF-based named entity extraction for Twitter Microposts. They developed NER for English and other three Indian languages including Malayalam and estimated an average precision for all those languages as part of FIRE 2015.

Another work was reported from Amrita University as part of FIRE 2014 [41]. In this paper, they evaluated the performance of different entity tagging algorithms in different languages. They used CRF for English and SVM for Indic languages. The size of the training corpus was very small. Lakshmi *et al.* [38] reported NER in Malayalam using fuzzy SVM. This system was based on contextual semantic rules and linguistic grammar rules. The system could solve ambiguity caused by traditional SVM classifier but the size of the corpus used was limited. Jisha P. Jayan *et al.* in 2013 [36] developed a NER for Malayalam in Hybrid Approach. They used TnT, an open-source statistical tagger for sequence labeling tasks. But the training corpus was so limited and they developed a NER in a closed domain. The first work in Malayalam was reported by Bindhu *et al.* in 2011 [39]. They used a combination of linguistic principles and statistical methods for NER. The limited size of the training corpus was a major drawback.

The proposed approach uses different DL architectures such as GRU, RNN, LSTM, BLSTM, etc., to improve the named entity recognition system performance. In this work, we use eight entity classes such as person, location, organization, miscellaneous, etc., and the entities not belonging to these classes will be arranged into another category.

3. A FRAMEWORK FOR NER FOR MALAYALAM USING DL APPROACHES

This section details the proposed approach for NER for the Malayalam language. This approach uses DL which is the current buzzword in the machine learning literature. While shallow-learning-based approaches use handcrafted features, DL approaches have already proved that it can better generate features thus improves learning for many areas where traditional machine learning approaches have degraded performances. The proposed method uses different DL approaches such as RNNs, GRU, LSTM, BLSTM as the network architectures for developing the named entity recognizer for Malayalam. With this, the authors would like to give a comparison of different DL architectures for the NER task for Malayalam. The overall workflow of the proposed approach is given in Figure 1.

In the following subsections, a brief introduction of the different DL architectures used in the proposed approach is detailed:

- **Recurrent Neural Networks**—This is an extension of a standard multi-layer perceptron that is able to manage the "recurrent states" to store sequences of variable length data. These states are interconnected for collecting the feedback and provide a simple way to work with sequential data [12,13]. In RNNs, the output of the states will be fed back to the same states which form a looping mechanism to collect the feedback. The computations at the hidden state ht are computed based on the current input xt and the previous hidden state ht-1. Even though the normal RNN is now widely used in many machine learning algorithms, it suffers from an issue called vanishing gradient. To overcome this, GRU is introduced.
- **Long Short-Term Memory**—These are considered as an extension of RNN but with a gating technique that can capture long dependencies. LSTM [13] can be seen as a memory cell that has two inputs—current input and the previous hidden state. The duty of the memory cell is to decide which information should be persisted and which should be removed. The more advanced variation of LSTM, known as BLSTM is also introduced which has the capability of capturing the sequence data in both forward and backward directions. Many machine learning and natural language processing applications that are operating on sequence data use this feature of BLSTM.
- **Gated Recurrent Unit**—This architecture uses a gating function in contrast with the LSTM and uses two types of gates for its operation—an update gate and a reset gate. The sigmoid activation function is computed using the very previous hidden state and current input which is taken by the reset gate.

The proposed work uses all these DL architectures—RNN, LSTM, BLSTM and GRU to work with the NER task. We use the publicly available labeled dataset for NER for Malayalam. The data is represented as a one-hot vector format and passed through the DL architectures discussed above. The aim of this approach is to compare the DL-based approaches with the simple neural network approaches. The tagset we have used for this experiment is shown in Table 1.

4. EXPERIMENTAL SETUP

4.1. Dataset

For this experiment, we have used the publicly available tagged NER dataset for Malayalam from https://cs.cusat.ac.in/mlpos/ner.zip. The dataset consists of 204833 entity tagged words and we have used 8 different tags for this experiment as shown in Table 1. A snapshot of the dataset we have used in this work is shown in Figure 2. In

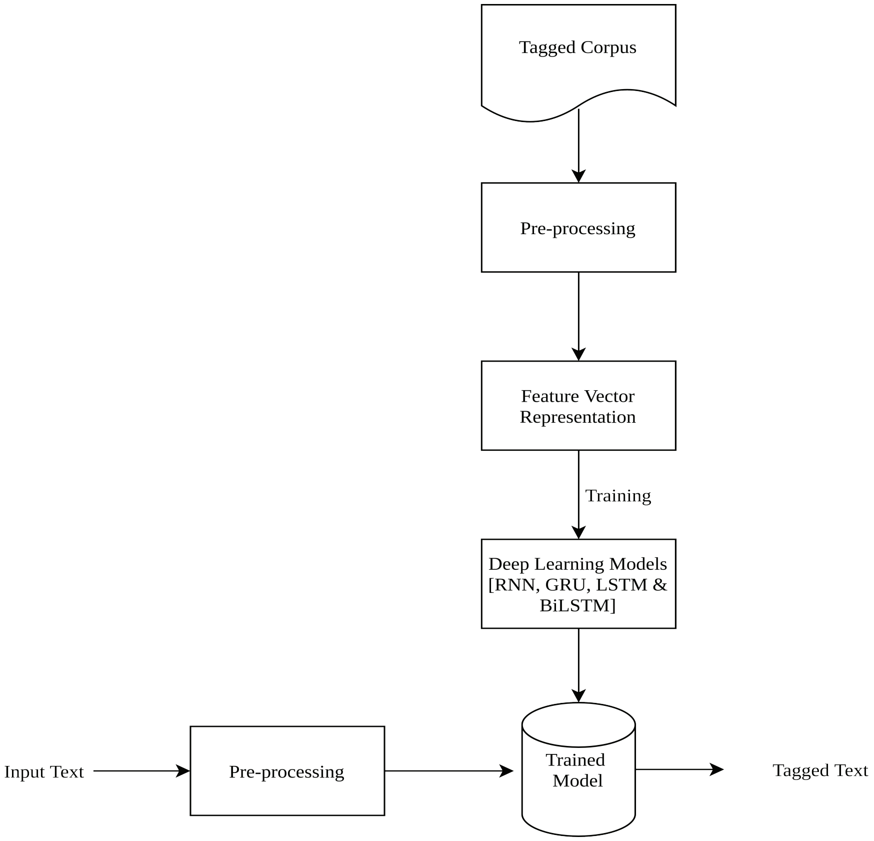


Figure 1 Overall workflow of the proposed method.

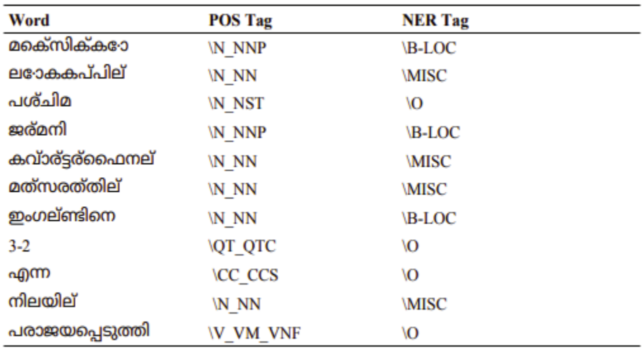
the figure, the first column denotes the tokenized word, second column represents the parts-of-speech tag, and in the third column we can see the named entity tag. While examining the dataset we have found that, the number of words tagged as “others” (denoted with “O”) and “miscellaneous” (denoted with “MISC”) are high. So to reduce the class imbalance issues while training, we have removed the words tagged as “others” and “miscellaneous” from the dataset.

4.2. Experiments

The experiments were implemented on a server computer that has a configuration of AMD Opteron 6376 @ 2.3 GHz and also has 16 core processors and 64 GB of main memory. We have used Python 3.7 with Keras machine learning toolkit for implementing the proposed DL methods such as LSTM, GRU, and BLSTM. We have experimented with GRU, RNN, LSTM, and Bi-LSTM architectures. We have chosen the hidden states as 4, 16, 32, and finally 64 as well. We have started with the number of iterations (epochs) as 30 and later increased the same to 50 and the last set of experiments used 100 epochs. For the models specified here, we have used the hidden layer size as 32 and the activation function is set as “tanh.” A dropout parameter is also used after many trial and error mechanisms to improve the training and the network used a learning parameter of 0.01. For this experiment, we have used evaluation measures such as precision, recall, and accuracy. Out of 204833 total tagged words available in the dataset, we have used 165000 for training and the rest for testing purposes.

Table 1 | The set of named entity tags used in the proposed work.

Entity Tag	Description
B-PER	Beginning of entity—Person
I-PER	Inside entity—Person
B-LOC	Beginning of entity—Location
I-LOC	Inside entity—Location
B-ORG	Beginning of entity—Organization
I-ORG	Inside entity—Organization
MISC	Miscellaneous
OUT	None of the above tags



Word	POS Tag	NER Tag
മെക്സിക്കോ	\N_NNP	\B-LOC
ലോകകപ്പില്	\N_NN	\MISC
പശ്ചിമ	\N_NST	\O
ജര്മനി	\N_NNP	\B-LOC
കവ്ാര്ട്ടര്ഫൈനല്	\N_NN	\MISC
മത്സരത്തില്	\N_NN	\MISC
ഇംഗല്ണ്ടിനെ	\N_NN	\B-LOC
3-2	\QT_QTC	\O
എന്ന	\CC_CCS	\O
നിലയില്	\N_NN	\MISC
പരാജയപ്പെടുത്തി	\V_VM_VNF	\O

Figure 2 | A snapshot of the dataset used for training.

4.3. Results and Evaluation

This section details the results of the experiment we have conducted and a detailed analysis of the same has also been given. For this experiment, we have done the performance evaluation for the four deep neural network architectures used - RNN, GRU, LSTM, and BLSTM. The four elementary matrices such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are used for the evaluation. TPs are the cases where the actual outcome is positive and the machine learning model also gave it as positive. TNs are cases anticipated appropriately as negative. FPs are cases anticipated as positive yet are negative cases. FNs are cases delegated negative yet are actually positive. Precision, recall, and accuracy is then calculated as given in Equations (1–3) respectively.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

F-measure is the harmonic mean of precision and recall and which can be computed using Equation (4).

$$F - measure = 2 * \frac{Precicion * Recall}{Precision + Recall} \tag{4}$$

The experimental results obtained using the experiment details given above in terms of precision, recall and F-measure are shown in Table 2. For the BLSTM, the parameter *input_dimension* was set as 1860, and *input_length* and *output_dimension* are given as 180. For the embedding layer, the dropout rate was set as 0.5 and We have used one layer of BLSTM with hidden units equal to 300. Number of layers in the feed forward network and number of cells in each layer is also one of the hyper parameters. This work used 3 dense layers, the first and second dense layers have 100 cells and the third layer having the number of cells equal to number of classes we need

Table 2 | Precision, recall and F-measure values of different deep learning approaches used.

Method	Precision	Recall	F-measure
RNN_with_4_layers	0.6389	0.6028	0.6023
RNN_with_16_layers	0.8122	0.7985	0.8052
RNN_with_32_layers	0.8223	0.8256	0.8239
RNN_with_64_layers	0.8380	0.8128	0.8252
GRU_with_4_layers	0.7122	0.7088	0.7104
GRU_with_16_layers	0.7363	0.7210	0.7285
GRU_with_32_layers	0.8258	0.8079	0.8167
GRU_with_64_layers	0.8399	0.8254	0.8325
LSTM_with_4_layers	0.7988	0.7844	0.7915
LSTM_with_16_layers	0.8356	0.8079	0.8215
LSTM_with_32_layers	0.8580	0.8450	0.8514
LSTM_with_64_layers	0.8752	0.8521	0.8634
BLSTM_with_4_layers	0.8597	0.8410	0.8502
BLSTM_with_16_layers	0.9144	0.8947	0.9044
BLSTM_with_32_layers	0.9354	0.9230	0.9291
BLSTM_with_64_layers	0.9541	0.9513	0.9526

Note: RNN, recurrent neural network; GRU, gated recurrent unit; LSTM, long short-term memory; BLSTM, bi-directional long short-term memory.

to categorize. The activation function used was *sigmoid*, the loss was *binary_crossentropy*, the optimizer was *adam* with a *binay* class mode. The batch size used are 1, 2, and 4 and from the mean performance, the results suggest lower root-mean-square error with a batch size of 1. The authors believe that this may be improved further with more training epochs.

From Table 2, it is evident that out of all the DL architectures we have used in this work, BLSTM shows better performance when trained with 64 layers. The precision, recall, and F-measure values are 95.41%, 95.13%, and 95.26% respectively. While RNN gave 83.80%, 81.28%, and 82.52% respectively for the precision, recall, and F-measure, the corresponding values for GRU was 83.99%, 82.54%, and 83.25%. The closest competitor for BLSTM is the LSTM based approach that showed 87.52%, 85.21% and 86.34% for precision, recall, and F-measure. To confirm that the DL-based methods outperform simple ANN-based approaches for the NER task, we have compared the performance of these DL architectures with the ANN-based approach. The experimental results show that in this proposed approach DL-based approaches showed better performance in terms of precision, recall, and F-measure. The performance comparison of different DL architectures used in this work is shown in Figure 3.

To further confirm the outperformance of DL-based approaches with the shallow-learning models, we have compared the top two best performers of the DL architectures—BLSTM and LSTM with the ANN approach on the same dataset. From the results, it is evident that DL-based approaches are performing significantly better than the ANN-based approach. The results are shown in Figure 4. The authors believe that it is worth mentioning the limitations and the issues faced in this research. As mentioned earlier, low-resource language computing heavily suffers from the unavailability of enough resources for the research and the same is with

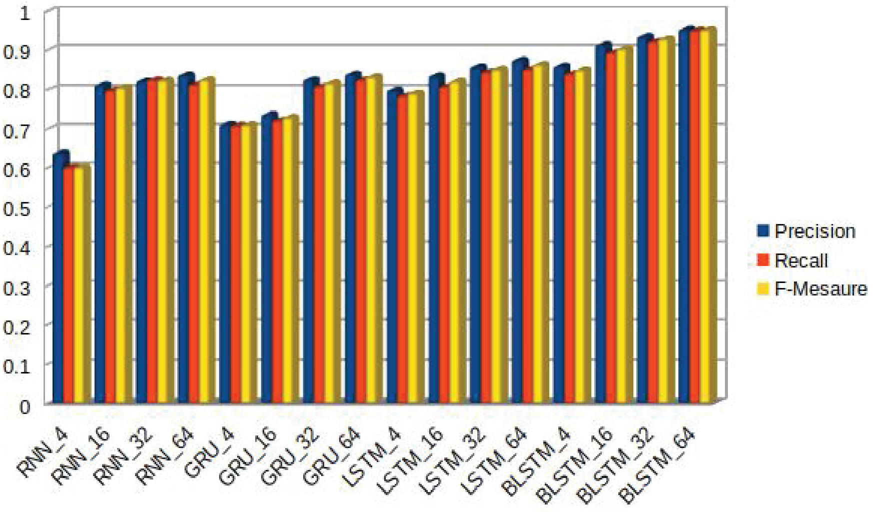


Figure 3 Performance comparison of different deep learning architectures for the named entity recognition (NER) task.

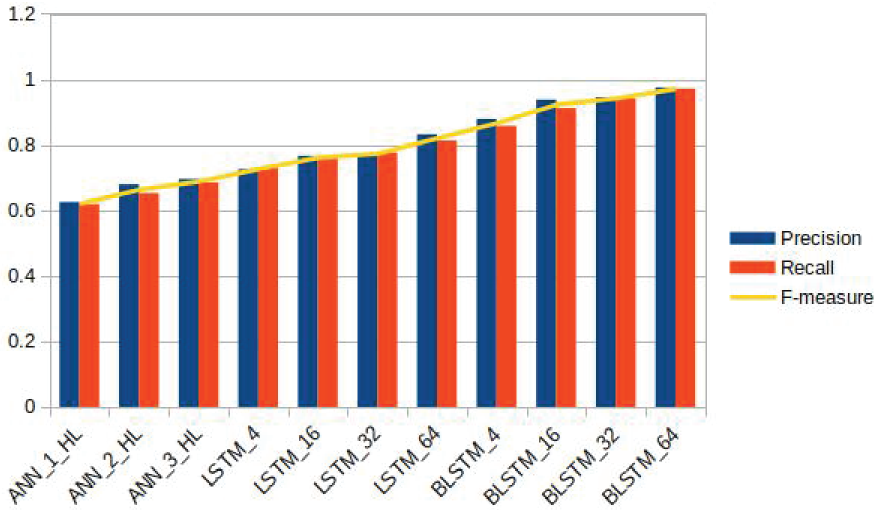


Figure 4 Performance comparison of long short-term memory (LSTM) and bi-directional long short-term memory (BLSTM) with artificial neural network (ANN).

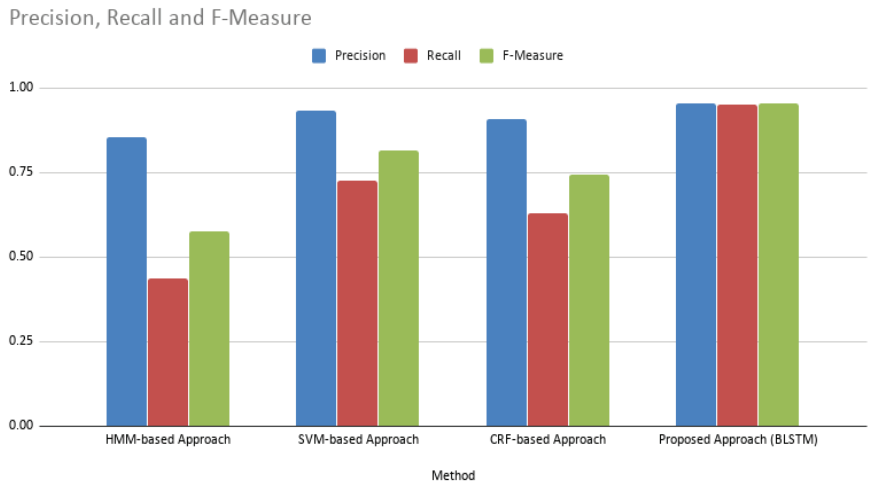


Figure 5 Performance comparison of our proposed named entity recognition (NER) approach with other state-of-the-art approaches.

Malayalam language computing as well. Even though the dataset we have used for this work is publicly available for research work, the data was not balanced. Thus many times the issue of overfitting happened with this experiment and we had to normalize the dataset. We had dropped some classes to make the experiment work. The community support and resources were also very limited so that when we stuck up with some issues, we had to figure out the same on our own. The authors believe that in the near future, more resources will be published and made available for low-resource language computing, specifically in Malayalam as we could see some serious efforts are being made among the researchers.

4.4. Comparison with the State-of-the-Art Malayalam NER Approaches

In this section, our proposed framework that uses DL for NER for Malayalam language is compared with some state-of-the-art approaches. We have chosen [35–37] as the baselines and evaluated the performance of our proposed DL-based approach with these approaches and found that the DL-based approach significantly outperforms the state-of-the-art approaches. The performance comparison is shown in Figure 5.

5. CONCLUSIONS AND FUTURE WORK

This paper proposed a DL-based approach for NER for Malayalam. The proposed method uses different DL approaches such as RNNs, GRU, LSTM, BLSTM architectures to implement and compare the performances. The experiments conducted using publicly available tagged entity corpus shows that DL approaches have significant potential and give better precision, recall, and accuracy over the shallow-learning-based approaches. This gives glimpses that many language computing algorithms developed for low-resource languages, specifically in Indic language communities, may give better accuracy if DL approaches can be implemented. But this poses many challenges, primarily the limited availability of tagged corpus and other resources to work with.

As the end results of this proposed work are promising, the authors would like to extend this experiment on a much larger dataset. Also, it is worth looking to tag more data and publish that for other low-resource language computing researchers. The authors of this proposed work also would like to develop a web application with these trained models to provide a named entity tagger for Malayalam.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHOR CONTRIBUTIONS

Anoop shaped the framework used in this article and provided sufficient information and guidance to the development. Rajimol has done the coding and analysis part and also collected the related literatures. Anoop and Rajimol contributed to the writing of the manuscript and Anoop has provided critical feedback and suggestive support to this work. All authors contributed to the final manuscript.

Funding Statement

The authors received no financial support for the research, authorship, and/or publication of this article.

ACKNOWLEDGMENTS

The authors would like to thank the staff members of Indian Institute of Information Technology and Management—Kerala for their support and facilities for completing this work. We are grateful to the anonymous reviewers for their valuable and constructional advices on the previous versions of this article; all remaining errors are our own.

REFERENCES

[1] D. Nadeau, S. Sekine, A survey of named entity recognition and classification, Lingvisticae Investig. 30 (2007), 3–26.

[2] G. Wei, Named entity recognition and an application to document clustering, M.Sc. Dissertation published in the year 2004 at Dalhousie University, Canada.1984.

[3] T. Wakao, R. Gaizauskas, Y. Wilks, Evaluation of an algorithm for the recognition and classification of proper names, in Proceedings of the 16th conference on Computational linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, 1996, vol. 1, pp. 418–423.

[4] D. Kaur, V. Gupta, A survey of named entity recognition in English and other Indian languages, Int. J. Comput. Sci. Issues. 7 (2010), 239.

[5] L. Zhang, Y. Pan, T. Zhang, Focused named entity recognition using machine learning, in Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, 2004, pp. 281–288.

[6] B. Mohit, R. Hwa, Syntax-based semi-supervised named entity tagging, in Proceedings of the ACL Interactive Poster and Demonstration Sessions, Stroudsburg, PA, USA, 2005, pp. 57–60.

[7] S. Amarappa, S.V. Sathyanarayana, Kannada Named Entity Recognition and Classification (NERC) based on Multinomial Naïve Bayes (MNB) classifier, arXiv preprint arXiv:1509.04385, 2015.

[8] H. Shabat, N. Omar, K. Rahem, Named entity recognition in crime using machine learning approach, in: A. Jaafar *et al*. (Eds.), Asia Information Retrieval Symposium, Springer, Cham, Switzerland, 2014, pp. 280–288.

[9] J.I. Kazama, T. Makino, Y. Ohta, J.I. Tsujii, Tuning support vector machines for biomedical named entity recognition, in Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, vol. 3, pp. 1–8.

[10] K. Takeuchi, N. Collier, Use of support vector machines in extended named entity recognition, in Proceedings of the 6th Conference on Natural Language Learning, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, vol. 20, pp. 1–7.

[11] H. Isozaki, H. Kazawa, Efficient support vector classifiers for named entity recognition, in Proceedings of the 19th International Conference on Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, pp. 1–7.

[12] J. Mayfield, P. McNamee, C. Piatko, Named entity recognition using hundreds of thousands of features, in Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, Association for Computational Linguistics, Stroudsburg, PA, USA, 2003, vol. 4, pp. 184–187.

[13] R. Arora, C.T. Tsai, K. Tsereteli, P. Kambadur, Y. Yang, A Semi-Markov structured support vector machine model for high-precision named entity recognition, in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019, pp. 5862–5866.

[14] K.K. Lay, A. Cho, Myanmar named entity recognition with hidden Markov model, Int. J. Trend Sci. Res. Dev. 3 (2019), 1144–1147.

[15] M.D. Drovo, M. Chowdhury, S.I. Uday, A.K. Das, Named entity recognition in Bengali text using merged hidden markov model and rule base approach, in 2019 7th International Conference on Smart Computing & Communications (ICSCC), IEEE, Sarawak, Malaysia, 2019, pp. 1–5.

[16] F. Alam, M.A. Islam, A proposed model for Bengali named entity recognition using maximum entropy Markov model incorporated with rich linguistic feature set, in Proceedings of the International Conference on Computing Advancements, Dhaka, Bangladesh, 2020, pp. 1–6.

[17] I.S. Azarine, M.A. Bijaksana, I. Asror, Named entity recognition on Indonesian tweets using hidden Markov model, in 2019 7th International Conference on Information and Communication Technology (ICoICT), IEEE, Kuala Lumpur, Malaysia, 2019, pp. 1–5.

[18] F. Dernoncourt, J.Y. Lee, P. Szolovits, NeuroNER: an easy-to-use program for named-entity recognition based on neural networks, arXiv preprint arXiv:1705.05487, 2017.

[19] J. Straková, M. Straka, J. Hajič, Neural networks for featureless named entity recognition in Czech, in: P. Sojka, A. Horák, I. Kopeček, K. Pala (Eds.), International Conference on Text, Speech, and Dialogue, Springer, Cham, Switzerland, 2016, pp. 173–181.

[20] N.F. Mohammed, N. Omar, Arabic named entity recognition using artificial neural network, J. Comput. Sci. 8 (2012), 1285.

[21] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, arXiv preprint arXiv:1603.01360, 2016.

[22] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, S. Khudanpur, Recurrent neural network based language model, in Eleventh Annual Conference of the International Speech Communication Association, Prague, Czech Republic, 2010.

[23] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, S. Khudanpur, Extensions of recurrent neural network language model, in 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Prague, Czech Republic, 2011, pp. 5528–5531.

[24] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555, 2014.

[25] R. Dey, F.M. Salemt, Gate-variants of Gated Recurrent Unit (GRU) neural networks, in 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE, Boston, MA, USA, 2017, pp. 1597–1600.

[26] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997), 1735–1780.

[27] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, arXiv preprint arXiv:1508.01991, 2015.

[28] G. Zhou, J. Su, Named entity recognition using an HMM-based chunk tagger, in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, pp. 473–480.

[29] S. Morwal, N. Jahan, D. Chopra, Named entity recognition using Hidden Markov Model (HMM), Int. J. Nat. Lang. Comput. 1 (2012), 15–23.

[30] H.L. Chieu, H.T. Ng, Named entity recognition: a maximum entropy approach using global information, in Proceedings of the 19th International Conference on Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, vol. 1, pp. 1–7.

[31] G. Paliouras, V. Karkaletsis, G. Petasis, C.D. Spyropoulos, Learning decision trees for named-entity recognition and classification,

in ECAI Workshop on Machine Learning for Information Extraction, Berlin, Germany, August, 2000.

[32] G. Szarvas, R. Farkas, A. Kocsor, A multilingual named entity recognition system using boosting and c4. 5 decision tree learning algorithms, in: L. Todorovski, N. Lavrač, K.P. Jantke (Eds.), International Conference on Discovery Science, Springer, Berlin, Heidelberg, Germany, 2006, pp. 267–278.

[33] A.P. Ajees, S.M. Idicula, A named entity recognition system for Malayalam using neural networks, Procedia Comput. Sci. 143 (2018), 962–969.

[34] A.P. Ajees, S.M. Idicula, A named entity recognition system for Malayalam using conditional random fields, in 2018 International Conference on Data Science and Engineering (ICDSE), IEEE, Kochi, India, 2018, pp. 1–5.

[35] G.R. Devi, P.V. Veena, M.A. Kumar, K.P. Soman, Entity extraction for Malayalam social media text using structured skip-gram based embedding features from unlabeled data, Procedia Comput. Sci. 93 (2016), 547–553.

[36] J.P. Jayan, R.R. Rajeev, E. Sherly, A hybrid statistical approach for named entity recognition for the Malayalam language, in Proceedings of the 11th Workshop on Asian Language Resources, Nagoya, Japan, 2013, pp. 58–63.

[37] G. Prasad, K.K. Fousiya, M.A. Kumar, K.P. Soman, Named entity recognition for the Malayalam language: a CRF based approach, in 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy, and Materials (ICSTM), IEEE, Chennai, India, 2015, pp. 16–19.

[38] G. Lakshmi, J.R. Panicker, M. Meera, Named entity recognition in Malayalam using fuzzy support vector machines, in 2016 International Conference on Information Science (ICIS), IEEE, Kochi, India, 2016, pp. 201–206.

[39] M.S. Bindu, S.M. Idicula, Named entity identifier for Malayalam using linguistic principles employing statistical methods, Int. J. Comput. Sci. Issues. 8 (2011), 185.

[40] M. Shruthi, A study on named entity recognition for Malayalam language using TnT tagger & maximum entropy Markov model, Int. J. Appl. Eng. Res. 11 (2016), 5425–5429.

[41] N. Abinaya, N. John, B.H. Ganesh, A.M. Kumar, K.P. Soman, AMRITA_CEN@ FIRE-2014: named entity recognition for Indian languages using rich features, in Proceedings of the Forum for Information Retrieval Evaluation, Bangalore, India, 2014, pp. 103–111.

[42] N. Patil, A.S. Patil, B.V. Pawar, Survey of named entity recognition systems with respect to Indian and foreign languages, Int. J. Comput. Appl. 134 (2016), 21–26. https://www.ijcaonline.org/archives/volume134/number16/23999-2016908197

[43] S.P. Sanjay, M.A. Kumar, K.P. Soman, AMRITA_CENNLP@ FIRE 2015: CRF based named entity extractor for Twitter microposts, in FIRE Workshops, Gandhinagar, India, December 4–6, 2015, pp. 96–99.

[44] Y. Gal, Z. Ghahramani, A theoretically grounded application of dropout in recurrent neural networks, in Advances in Neural Information Processing Systems, Centre Convencions Internacional Barcelona, Barcelona SPAIN, 2016, pp. 1019–1027.

[45] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997), 1735–1780.

[46] A. Akbik, T. Bergmann, R. Vollgraf, Pooled contextualized embeddings for named entity recognition, in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2019, vol. 1, pp. 724–728.

[47] C. Sabty, M. Elmahdy, S. Abdennadher, Named entity recognition on Arabic-English code-mixed data, in 2019 IEEE 13th International Conference on Semantic Computing (ICSC), IEEE, Newport Beach, CA, USA, 2019, pp. 93–97.

[48] G.I. Winata, Z. Lin, P. Fung, Learning multilingual meta-embeddings for code-switching named entity recognition, in Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019), Florence, Italy, 2019, pp. 181–186.

[49] A. Johnson, P. Karanasou, J. Gaspers, D. Klakow, Cross-lingual transfer learning for Japanese named entity recognition, in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Industry Papers), Minneapolis, MN, USA, 2019, vol. 2, pp. 182–189.

[50] P.S. Sreeja, A.S. Pillai, Towards an efficient Malayalam named entity recognizer analysis on the challenges, Procedia Comput. Sci. 171 (2020), 2541–2546.