GRU based Named Entity Recognition System for Bangla Online Newspapers

Conference Paper · December 2018

DOI: 10.1109/CIET.2018.8660795

Nayan Banik
Comilla University

Md. Hasan Hafizur Rahman
Comilla University

GRU based Named Entity Recognition System for Bangla Online Newspapers

Nayan Banik
Department of Computer Science & Engineering
Comilla University
Comilla - 3506, Bangladesh
Email: cse.nayan@gmail.com

Md. Hasan Hafizur Rahman
Department of Computer Science & Engineering
Comilla University
Comilla - 3506, Bangladesh
Email: hhr@cou.ac.bd

***Abstract*—Information Extraction (IE) from textual documents locates important entities and their underlying connections using automated systems which are crucial to different applications including Data Mining (DM), Question Answering (QA), Machine Translation (MT) and so on. Named Entity Recognition (NER) being a sub-component of Natural Language Processing (NLP) is an IE task which aims at locating the textual presence of entities belonging to a prescribed set of classes. Due to its political and geographical influence, Bangla language is widely spoken around the globe and it is important to enrich its linguistic knowledge through NLP tools where NER is a common pre-processing step. The expeditiously growing World Wide Web (WWW) containing Bangla textual documents is in a formative stage with the proliferation of Bangla online newspapers and researchers have applied traditional classic learning algorithms for Bangla NER task while few researchers have used hand-crafted rules. Technological improvements show that with the capability of Deep Learning technique, NER performance can be boosted and hence this work is an effort to apply a variation of Recurrent Neural Network (RNN); especially a Gated Recurrent Unit (GRU) model for developing a Bangla NER task with a manually annotated dataset. The evaluation of our experimental results discovers how our approach can perform better when applied on a large scale dataset.**

Keywords*—*Named Entity Recognition*, *Information Extraction*, *Natural Language Processing

I. Introduction

A newspaper is a great source of structured information ranging from many categories where readers can get the latest updated news in their preferred categories. Aside from traditional print-based newspaper media, digitalized online news portals are getting popular day-by-day due to the widespread use of the Internet and the presence of online users. The manifold benefits of online newspapers from the reader's perspective and editor's perspectives can easily be understood since the portals get huge exposure on the Internet and the readers can get the latest news update as it is happening. Considering these benefits, online newspapers are increasing rapidly and this vast source of information needs to be processed for both human and machine consumption.

$Bangla$ language ranks seventh according to the number of speakers around the globe[1]. And as a South Asian language and the mother tongue of Bangladesh, it is widely used for day-to-day communication among the peoples around the world for its importance in geographical and political purposes. Hence Bangla online newspapers providing news on different topics are now shared continuously among the readers through Internet and it is predicted that it will increase in upcoming years. These textual documents are a great source of information for both human and machines to take decisions and to make automated systems.

Natural Language Processing (NLP) aims at extracting important features from language data i.e. speech and text using available computing resources and this computational field requires digitized annotated language data in a structured form to provide information ready for human and machine consumption. Though previously extracting information from textual data was hard due to the lack of specialized tools and automated processing, the technological revolution has overcome this scenario with cheap computers having adequate capacity to process data in a standardized way [2].

Identification of proper nouns like *Person*, *Location*, *Organization*, *Time*, *Number* etc in a textual document is a major task in NLP. These words having rigid designation is called Named Entities (NE) and the process to classify them is called Named Entity Recognition (NER) [3]. This process is a required for different NLP based applications including IE, MT, QA, text summarization, search results clustering etc. The major two approaches for performing the NER includes manual handcrafted rule-based techniques and statistical feature extraction using machine learning based techniques. The manual handcrafted rule-based techniques usually map linguistic grammar to the regular expression and it requires expert linguist to develop the rules which can take a long time [4][5]. In statistical techniques, the NER can be modeled as a multi-class classification problem and various crucial characteristics are learned to classify the entities where the language independent tagging is used [6]. For a scarcely inflected language like Bangla, having a comparatively large expert data for training is a limiting factor in the statistical machine learning based system.

In order to explore Deep Learning techniques for NER system in Bangla, we have used Gated Recurrent Unit (GRU); a variation of Recurrent Neural Network (RNN) model that utilize manually annotated Bangla online newspapers dataset. To experience that, we manually collected data from a popular online Bangla newspaper and apply necessary pre-processing

978-1-5386-5229-9/18/$31.00 ©2018 IEEE

before feeding to the network. For the training purposes, we use publicly available python libraries. The experimental outcomes demonstrated that our developed model achieved F1-score of 69%.

This research paper is documented as given. In **Section II**, related works on Bangla NER is discussed. **Section III** describes our approach and methodology. **Section IV** discusses our implementation procedures. System performances and associated evaluatory analysis are described in **Section IV** and the paper concludes with the conclusion in **Section V**.

II. Related Works

Due to its complicated morphological formation and free-form complex sentence structures, Bangla language lacks required NLP tools for doing research in this linguistic domain[7][8]. In order to tackle these issues, many researchers have done their research on Bangla NER task like in [9], authors propose an algorithm using the partial string matching technique for Breadth First Search (BFS) on a Trie data structure. They claim to detect NE on unstructured Bangla text from online newspapers but since they apply their approach to a closed domain having a predefined list of NE, the system is limited on that particular domain. Another work in [10], authors discuss how Word Embedding (WE) can be helpful to learn word vectors for low resource language like Bangla so perform NER task. They demonstrate their approach having overall F-score of 65.4% where the traditional Conditional Random Field (CRF) based system has 64.2%. Multilingual NER system utilizing Multi Objective Optimization (MOO) technique is done in [11] [12][13][14]. In these works, the authors use MOO for extracting features and optimizing the required parameters. They also demonstrate how performance can be improved for low resource language like Bangla by utilizing classifier ensemble technique over single classifier. In [15], the authors develop a biomedical NER system for Bangla and Hindi by applying Support Vector Machines (SVM) and ensemble methods. They use active learning concept to extract representative information from unannotated textual documents. The resulting performance depicts how the traditional approach can be improved through annotation by a high margin and for Bangla, it achieved F-measure of 87.94%. Relating to that, cross-lingual NER on different languages including Bangla proposed in [16] discusses how Wikipedia data can be utilized to perform language independent NER task. In this work, the authors claim that this model would work on all languages in Wikipedia and the quantity of test language has a direct connection with the representative quality of important features. Margin Infused Relaxed Algorithm for Bangla NER task is proposed in [17] where the authors consider both language dependent and independent features and the system outperforms baselines with precision 91.23%, recall 87.29% and F-measure 89.69% respectively when applied to a publicly available annotated corpus with twelve NE tagsets. CRF based Bangla NER system in [18] authors consider language independent features on a small dataset and using gazetteers, the system achieves F-measure of 87%. Hidden Markov Model (HMM) based NER system in [19] achieves experimental F-measure of 0.8599. A voted NER system exploiting unlabeled data using several machine learning algorithms like Maximum Entropy (ME), CRF and SVM is proposed in [20] where the authors develop a combined system by incorporating separate individual classifier results through weighted voting technique. Considering this, the experimental results achieve F-measure of 92.98%. Another work in the tourism domain proposed in [21], the authors use HMM and they claim that the result having F-measure of 98%. The authors in [22] claim that unavailability of expert data and manual preparation of them poses a challenge because it requires costly resources to annotate. According to the authors, this limitations can be overcome by ensemble based active annotation technique and they demonstrate it by combining two supervised classifiers, namely SVM and CRF achieving F-measure of 88.71%. In [23], the authors demonstrate different feature reduction techniques for improving the performance of Bangla NER system. Relating to that in [24], a two-step evolutionary technique for developing a NER system for Bangla is done. In the first phase, CRF and SVM is used for selecting representative features and in second phase those classifiers are ensembled using Differential Evolution (DE). Similar to that, CRF based Bangla NER done in [25] for a shared task results in performance of 81.15%. ME based Bangla NER system developed in [26] considers both language dependent and independent features resulting in F-score of 85.22%. In order to extract suitable feature combination for developing a Bangla NER system through Genetic Algorithm (GA) is proposed in [27]. In this research work, the authors apply ME based classifier and the system provides F-measure of 77.09%. In [28], authors have explored that Wikipedia-related features significantly improve a baseline Bangla NER system. Comprehensive states on earlier Bangla NER system is also discussed in [29][30][31][32].

III. Methodology

Recurrent Neural Network (RNN) [33] as a sequential data processor is used to repeatedly apply the same operations on each entity in a series where each individual activity depends on the previous operational outputs. The sequence is represented by a fixed-size vector which is feed to the recurrent unit one by one. The property of using previous results in present processing models a "memory" component in RNN and it is generally suited for many NLP tasks [34].

The simple recurrent unit also known as the Elman unit [33] is a three-layer network and in Fig. 1 a more general unfolded RNN across time representing a whole sequence is shown. According to Fig. 1,

- For a particular instant of time t, the input is denoted by x_t. Considering our NER system, it consists of one-hot encoding or embedding of the vocabulary.
- For the same time step t, hidden state is represented by s_t and it is generated by the equation:

$$s_t = f(Mx_t + Wds_{t-1})$$

where the hidden state value depends on the present input value and past hidden state value.

- The activation function f represent any non-linearities e.g. sigmoid, ReLU, tanh etc.
- Shared weights in the network are represented by M, N, Wd.
- The output of the network is denoted by Out_t and for a multi-network, it represents a non-linearity.

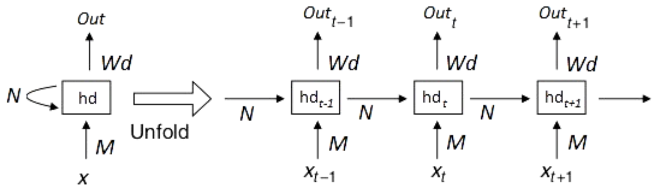

Fig. 1. Simple RNN

An important element of the RNN is the hidden state and it represents the memory element that stores values from previous time steps. Simple RNN networks suffer from the vanishing gradient problem and as a result earlier layers parameters are hard to learn and tune. In order to tackle this limitations, various networks are used such as Long Short Term Memory ($LSTM$), Gated Recurrent Units (GRUs), and Residual Networks ($ResNets$). But for NLP based applications e.g. NER system, $LSTM$ and GRU are widely used.

Unlike simple RNN, Long Short-Term Memory ($LSTM$) [35][36] comprises of additional ”forget” gates to allow the error in the network to back-propagate for an unlimited number of time steps as shown in Fig. 2. Combination of three gates (input, forget and output) as per the equations below, the $LSTM$ unit calculates the values for hidden states.

$$x = \begin{bmatrix} hd_{t-1} \\ x_t \end{bmatrix}$$

$$k_t = \sigma(Wd_k{\cdot}x + b_k)$$

$$j_t = \sigma(Wd_j{\cdot}x + b_j)$$

$$l_t = \sigma(Wd_l{\cdot}x + b_l)$$

$$Co_t = k_t{\odot}Co_{t-1} + j_t{\odot}\tanh(Wd_{Co}{\cdot}x + b_{Co})$$

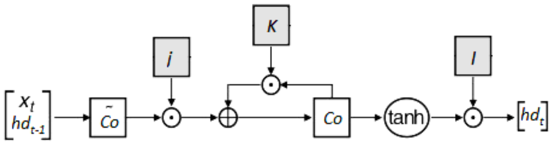

Fig. 2. Illustration of an LSTM

Gated Recurrent Unit (GRU) [37] is another variant of gated RNN as shown in Fig. 3 which is less complex than $LSTM$ but provides somewhat similar performance for NLP tasks. Similar to $LSTM$, it circulates the data flow through reset gate and update gate. But since it lacks any memory component, the overall structure is exposed.

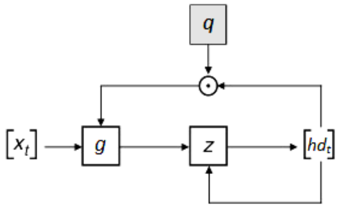

Fig. 3. Illustration of an GRU

Though the applicability of $LSTM$ or GRU for solving problems depends on heuristics, research work claims that GRU is effective and perform better than $LSTM$ [34]. The inner working of a GRU is provided below:

$$z = \sigma(M_z{\cdot}x_t + Wd_z{\cdot}hd_{t-1})$$

$$q = \sigma(M_q{\cdot}x_t + Wd_r{\cdot}hd_{t-1})$$

$$g_t = tanh(M_z{\cdot}x_t + Wd_s{\cdot}(hd_{t-1}{\cdot}q))$$

$$hd_t = (1-z){\odot}g_t + z{\odot}hd_{t-1}$$

IV. Implementation

Our proposed Bangla NER system is built using a supervised machine learning approach and in order to develop it, we have collected the raw data from a reputed source and manually annotated them. Then we have removed noise and performed tokenization by applying some pre-processing steps. Before feeding the data to the network, we have prepared them in standard form and finally generated the performance report. The overall system structure is depicted in Fig 4 and the whole process is described accordingly.

A. Data Collection

Our proposed NER system requires annotated Bangla online newspapers dataset which is currently not publicly available. In order to develop, train and evaluate our system, we have collected data from a reputed online Bangla newspaper ”*Prothom Alo*[1]” which is the most read news portals among the Bangla speakers around the globe. For this purpose, we have developed a custom crawler using *BeautifulSoup*[2]; a fast, simple and yet extensible python library for extracting and parsing structured information from any website. Among many news categories, we have considered four popular news categories named National, International, Entertainment, Technology and Sports. The dataset statistics are described in Table I.

[1] https://www.prothomalo.com

[2] https://www.crummy.com/software/BeautifulSoup/

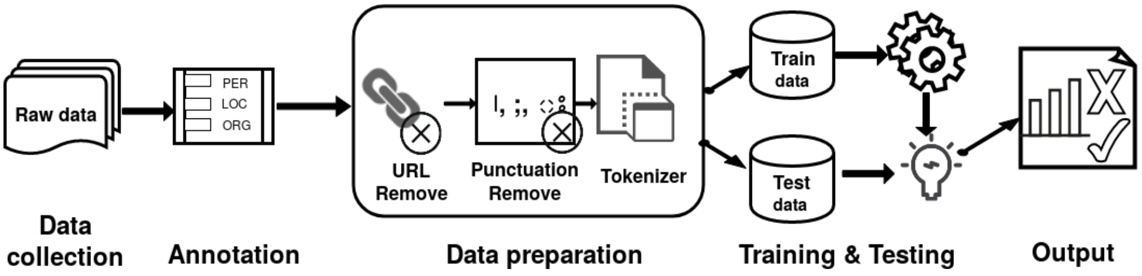

Fig. 4. System Structure

TABLE I
DATASET STATISTICS

Category	**Articles Count**
National	100
International	120
Entertainment	70
Technology	55
Sports	75
Total	420

B. Data Annotation

Our system utilized supervised approach for NER and hence we have annotated our collected data by two independent annotators. Among many standard NE, we have considered four categories which are very common in Bangla online newspapers. They NE are Person (the human subject of the news document), Location (any specific instance of a place including country, city, state etc), Organization (having independent identity among national and international perspectives) and day (very common in news articles). Annotators are guided to use IOB format for tagging the NE where B indicates whether the tag is the start of a NE, I indicates whether the tag is inside and O indicates not a NE. For example, *B-ORG-Apple CEO B-PER-Tim I-PER-Cook introduces IPhones at B-LOC-New I-LOC-York B-DAY-sunday*. The detailed NE tags in our annotated dataset are described in Table II.

TABLE II
NE TAGS COUNT IN DATASET

NE	**Tag Counts**
Person	1306
Location	1339
Organization	572
Day	183

C. Data Preparation

In our NER system and in most of the NLP systems, data pre-processing mainly aims to simplify the input and reduce the feature space in a manner that is suitable for the system to process. So we have implemented some of text pre-processing directly over the annotated raw data.

URL removal: Many news texts contains links to related information and since these URLs do not convey any NE we removed them in the first place.

Punctuation character removal: As we consider each word and its NE separated by a space on a line and no punctuation characters denotes any NE, we have removed them.

Text Segmentation: Separating each meaningful entity like words, phrase , sentence is called text segmentation. In our case, because we need to separate each words in a sentence and its associated NE tags one in a line, we have used native python string split functions to tokenize the words in sentences.

D. Training and Testing

Similar to all supervised machine learning approach, we have split our dataset into train data and test data. The model is then trained with the training data to learn the associated parameters of the network. The test data is then fed to the model to evaluate the performance. Due to the limited size of dataset, validation data is not used here. Since the system considers language independent annotation scheme, K-fold validation can be applied for increasing the performance with a moderate dataset in this domain.

E. Libraries

Different popular python based machine learning libraries are used to train and test our network. These libraries are also publicly available to use. Among them, *TensorFlow*[3] is an open source library for numerical computation and large-scale machine learning and it bundles together a bunch of machine learning and deep learning models and algorithms that makes them useful by way of a common interface. It works a backend to our network. *Theano*[4] is another python library that allows users to compute and optimize complex mathematical computations relating high-dimensional arrays. Just like TensorFlow, it is also used as a backend to improve our background computation of our network. *Scikit-Learn*[5] is also a python library which has clean, uniform, and stream-lined API and provides solid implementations of a range of machine learning algorithms. We have also employed it to process and streamline our network.

[3] https://www.tensorflow.org/
[4] http://deeplearning.net/software/theano/
[5] http://scikit-learn.org/stable/

V. Experimental Results and Evaluations

From several evaluation metrics, we have considered F1-score (or f-measure); the harmonic mean of precision and recall.

$$F1 - score = 2 \times \frac{P \times R}{P + R}$$

where P stands for precision and R for recall. The performance statistics of our proposed system is provided in Table III and the corresponding trends is visualized in Fig. 5.

TABLE III
Performance statistics

Vocabulary size	23137
No. of classes	8
Iteraions	100
Epoch	50/iteration
Learning rate	1e-4
Hidden layers	10
Activation function	ReLU
Recurrent unit	GRU
Training accuracy	0.99764
Testing accuracy	0.93311
Training F1-score	0.98262
Testing F1-score	**0.69420**

The relation between training cost and correct rate of our NER system is shown in Fig. 5 where it depicts how the network starts at a high cost and having a poor correct rate. The upper slope indicating the cost decrease indicates that hidden layers are adjusting the weights for learning the underlying relations among the NE with the surrounding contextual words. Since we applied only word indexing through dictionary lookup, the network requires comparatively more time to converge than using a pre-trained word-embeddings model. It can also be claimed that since the number of hidden layers is considerably small for our limited dataset, increasing it can results in better performance when more data can be incorporated. We have used ReLU as activation function because it is less prone to noise and its learning ratio aligned to our dataset. Though tanh also performs well, it takes a lot of time to converge. We use GRU as a recurrent unit for its benefits over $LSTM$.

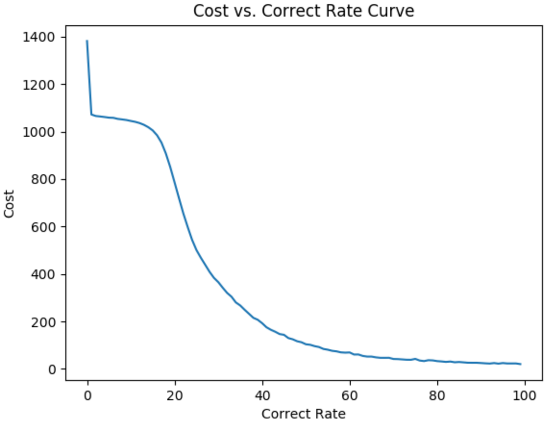

Fig. 5. Cost vs. Correct rate of the NER system

VI. Conclusion

Named Entity Recognition (NER) system as a sub-component of different Natural Language Processing (NLP) applications has a crucial role to extract entities of predefined categories from textual documents. NER system for $Bangla$ language using rule-based and statistical machine learning approach has been done for many domains. In this research work, a Gated Recurrent Unit (GRU) based NER system to identify four named entities (Person, Organization, Location, Day) from a reputed Bangla online newspaper is developed where the system utilizes language independent features on a manually annotated dataset. For the limited dataset, the system provides F1-score of 69%. The evaluative results also depict that the model can perform better by incorporating more data and using pre-trained word embeddings.

References

[1] "The 10 most spoken languages in the world," https://www.babbel.com/en/magazine/the-10-most-spoken-languages-in-the-world/, (Accessed on 11/20/2018).

[2] A. Goyal, V. Gupta, and M. Kumar, "Recent named entity recognition and classification techniques: A systematic review," *Computer Science Review*, vol. 29, pp. 21–43, 2018.

[3] R. Grishman and B. Sundheim, "Message understanding conference-6: A brief history," in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, vol. 1, 1996.

[4] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[5] S. Sarawagi *et al.*, "Information extraction," *Foundations and Trends® in Databases*, vol. 1, no. 3, pp. 261–377, 2008.

[6] M. Bhagavatula, S. GSK, and V. Varma, "Named entity recognition an aid to improve multilingual entity filling in language-independent approach," in *Proceedings of the first workshop on Information and knowledge management for developing region*. ACM, 2012, pp. 3–10.

[7] M. Karim, *Technical challenges and design issues in bangla language processing*. IGI Global, 2013.

[8] F. Alam, S. Habib, D. A. Sultana, and M. Khan, "Development of annotated bangla speech corpora," 2010.

[9] N. Ibtehaz and A. Satter, "A partial string matching approach for named entity recognition in unstructured bengali data," *International Journal of Modern Education and Computer Science*, vol. 10, no. 1, p. 36, 2018.

[10] A. Das, D. Ganguly, and U. Garain, "Named entity recognition with word embeddings and wikipedia categories for a low-resource language," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 16, no. 3, p. 18, 2017.

[11] A. Ekbal and S. Saha, "A multiobjective simulated annealing approach for classifier ensemble: Named entity recognition in indian languages as case studies," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14 760–14 772, 2011.

[12] ——, "Multiobjective optimization for classifier ensemble and feature selection: an application to named entity recognition," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 15, no. 2, pp. 143–166, 2012.

[13] ——, "Simultaneous feature and parameter selection using multiobjective optimization: application to named entity recognition," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 4, pp. 597–611, 2016.

[14] ——, "Combining feature selection and classifier ensemble using a multiobjective simulated annealing approach: application to named entity recognition," *Soft Computing*, vol. 17, no. 1, pp. 1–16, 2013.

[15] A. Ekbal, S. Saha, and U. K. Sikdar, "On active annotation for named entity recognition," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 4, pp. 623–640, 2016.

[16] C.-T. Tsai, S. Mayhew, and D. Roth, “Cross-lingual named entity recognition via wikification,” in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 219–228.

[17] S. Banerjee, S. K. Naskar, and S. Bandyopadhyay, “Bengali named entity recognition using margin infused relaxed algorithm,” in *International Conference on Text, Speech, and Dialogue*. Springer, 2014, pp. 125–132.

[18] A. Das and U. Garain, “Crf-based named entity recognition@ icon 2013,” *arXiv preprint arXiv:1409.8008*, 2014.

[19] V. Gayen and K. Sarkar, “An hmm based named entity recognition system for indian languages: the ju system at icon 2013,” *arXiv preprint arXiv:1405.7397*, 2014.

[20] A. Ekbal and S. Bandyopadhyay, “Named entity recognition in bengali using system combination,” *Lingvisticæ Investigationes*, vol. 37, no. 1, pp. 1–22, 2014.

[21] S. Morwal and D. Chopra, “Identification and classification of named entities in indian languages,” *International Journal on Natural Language Computing (IJNLC) Vol*, vol. 2, pp. 37–43, 2013.

[22] A. Ekbal, S. Saha, and D. Singh, “Ensemble based active annotation for named entity recognition,” in *Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on*. IEEE, 2012, pp. 331–334.

[23] S. K. Saha, P. Mitra, and S. Sarkar, “A comparative study on feature reduction approaches in hindi and bengali named entity recognition,” *Knowledge-Based Systems*, vol. 27, pp. 322–332, 2012.

[24] U. K. Sikdar, A. Ekbal, and S. Saha, “Differential evolution based feature selection and classifier ensemble for named entity recognition,” *Proceedings of COLING 2012*, pp. 2475–2490, 2012.

[25] A. Ekbal and S. Bandyopadhyay, “A conditional random field approach for named entity recognition in bengali and hindi,” *Linguistic Issues in Language Technology*, vol. 2, no. 1, pp. 1–44, 2009.

[26] M. Hasanuzzaman, A. Ekbal, and S. Bandyopadhyay, “Maximum entropy approach for named entity recognition in bengali and hindi,” *International Journal of Recent Trends in Engineering*, vol. 1, no. 1, p. 408, 2009.

[27] M. Hasanuzzaman, S. Saha, and A. Ekbal, “Feature subset selection using genetic algorithm for named entity recognition,” in *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*, 2010.

[28] K. S. Hasan, V. Ng *et al.*, “Learning-based named entity recognition for morphologically-rich, resource-scarce languages,” in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009, pp. 354–362.

[29] D. N. Shah and H. Bhadka, “A survey on various approach used in named entity recognition for indian languages,” *International Journal of Computer Applications*, vol. 167, no. 1, 2017.

[30] S. Kale and S. Govilkar, “Survey of named entity recognition techniques for various indian regional languages,” *International Journal of Computer Applications*, vol. 164, no. 4, 2017.

[31] H. Shah, P. Bhandari, K. Mistry, S. Thakor, M. Patel, and K. Ahir, “Study of named entity recognition for indian languages,” *Int. J. Inf*, vol. 6, no. 1, pp. 11–25, 2016.

[32] N. Patil, A. S. Patil, and B. Pawar, “Survey of named entity recognition systems with respect to indian and foreign languages,” *International Journal of Computer Applications*, vol. 134, no. 16, 2016.

[33] J. L. Elman, “Finding structure in time,” *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[34] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” *arXiv preprint arXiv:1708.02709*, 2017.

[35] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[36] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with lstm,” 1999.

[37] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.