End to End Parts of Speech Tagging and Named Entity Recognition in Bangla Language

Jillur Rahman Saurav
Shahjalal University of Science and Technology
1 PUBLICATION **0** CITATIONS
SEE PROFILE

End to End Parts of Speech Tagging and Named Entity Recognition in Bangla Language

Jillur Rahman Saurav*, Summit Haque†, Farida Chowdhury*
* Search Engine Pipilika
*†Department of Computer Science & Engineering
*†Shahjalal University of Science & Technology
Sylhet, Bangladesh
{sauravsust71, summit.haque, deeba.bd}@gmail.com

***Abstract*—Automatic Parts of Speech(POS) tagging is one of the most fundamental tasks for a language in Natural Language Processing(NLP), which acts as a feature for solving advanced NLP tasks. Named Entity Recognition(NER) is another essential task of NLP for information retrieval. Researchers could not find up to the mark solution yet on these two tasks for Bangla language compared to other languages, for instance, English, German. Moreover, many solutions heavily depend on handcrafted features that require strong linguistic expertise. As these two sequence labeling tasks are similar, In this work, two different datasets of POS tagging and NER were prepared, and different deep neural network approaches studied for solving these two tasks separately. All of the approaches were end to end and did not need any handcrafted feature like word suffixes or affixes, gazetteers, dictionary. This study came up with an end to end solution using deep neural network-based model consisting of Bi-directional Long short-term memory(BLSTM), Convolutional Neural Network(CNN) and Conditional Random Field(CRF). The proposed model trained on respected datasets achieved an accuracy of 93.86% on POS tagging and a strict f1 score of 0.6285 on NER on prepared datasets, respectively.**

***Index Terms*—Parts of Speech tagging, Named entity recognition, Bangla POS tagging, Bangla NER, Deep neural network, CNN, LSTM, BLSTM, CRF**

I. Introduction

In this era of technology, using NLP and Computer vision, machines are taught to mimic humans on different tasks. Even machines are now playing the role of the personal assistant. Different approaches are studied to make a machine capable of understanding human behavior, interactions.

One goal of NLP is to help the machine to understand human language and respond accordingly naturally. There are some fundamental things of a language, machines needed to learn for developing more advance Artifical Intelligent based system in that language. Extracted POS from given text is one of them. Recognizing a named entity is another necessary task of NLP to perform information retrieval. Solving these kinds of fundamental tasks will help to develop a more advanced system like a chatbot, optimize search results, etc.

In resource-rich languages on the perspective of NLP, researchers have made significant improvements in solving those tasks. For POS tagging in the English language, they have achieved over 97% accuracy [1].

Such kind of benchmark works on primary tasks of NLP has not been found for Bangla, the 7th most spoken language of the world [2].

In this work, we have experimented different approaches for automatic POS tagging and NER in Bangla language. These two tasks are similar on the perspective of machine because it needs to predict a tag for each token of a given text whether a POS tag or NER tag. We have studied end to end approaches where handcrafted features are not necessary during training and serving. We have explored different Deep Neural network models(DNN) as these can learn the non-linearities and now widely been used for almost all pattern recognition and machine learning application and achieving state-of-the-art performances in many sectors [3]. We have used word embeddings as in recent, NLP researchers found that using word embeddings or vector representation of words brings significant increases in the performance [4]. In recent studies, it is found that many problems can be solved by end to end approaches with more data and complex deep neural networks, for instance, end to end speech recognition [5] that motivated us to try end to end approach for these tasks.

Like other resourceful languages, for Bangla language, we can not find so many publicly available datasets to study. So we have prepared our datasets maintaining standards for conducting our research. The making of datasets is briefly described in Section III dataset preparations.

We came up with a model based on BLSTM, CNN and CRF, trained and tested respectively on the two datasets we have prepared for two tasks, outperformed other models on both POS tagging and NER.

The rest of the paper is organized as follows. Section II describes the related work. Section III presents the Datasets used

978-1-7281-5242-4/19 ©2019 IEEE

in this work. Section IV explains the deep neural networks that we have used in this work. Section V describes the training process and hyperparameters. The results are discussed in section VI, and we conclude in sectionVII

II. Related Work

More works have been done on POS tagging compared to NER in Bangla language. Some notable previous works on NER and POS tagging are presented below.

A. Named Entity Recognition

In Bangla language, most of the works were done by Ekbal et al. [6]- [13]. They showed the use of CRFs, Maximum Entropy(ME), Support Vector Machine(SVM)s and achieved f1 measure varies from 82% to 91% on the different number of entity types. A resource-based study was done by Chaudhuri et al. [14] using a dictionary, rules, and n-gram based statistical modeling. The reported accuracy by K. S. Hasan et al. is 71.9% on three entity types. The baseline model used by them was CRF.

B. POS recognition

Many experiments have been conducted in the area of POS recognition in Bangla language. These studies attempted to define tag sets and application of different statistical and machine learning model for automatic POS tagging.

CRF based system is featured with word suffixes with lexicons, and NER was proposed by this work [16] for pos tagging. They use 26 tags and a corpus containing 72,341 tokens and achieved 90.3 % accuracy. Another study was done on the same corpus with the Hidden Markov model(HMM) and ME based models [17]. They showed that the ME model beats the HMM model by a margin of 7.5 % more accuracy achieving 88.1% accuracy. In [18], they applied SVM and showed that SVM outperformed the previous CRFs, HMMS, and ME based system for the same corpus. An unsupervised approach for recognizing POS in resource-scarce language was proposed by [19]. For a dataset containing ten tags, they achieved an F1 score of 79 %.

In [20], the authors studied different algorithm on a corpus containing 4000 sentences(tagset not sure). They showed that Global Linear Model (GLM) outperformed HMMs, SVM, CRFs, ME obtaining an accuracy of 93.12%.

A deep neural network approach was proposed by this work [21]. They used Bi-directional LSTM based model with CRF layer on top. They got an accuracy of 86% on the dataset prepared by this work [22].

III. Dataset Preparation

A. NER Dataset

We have prepared our own NER dataset for this research. We have collected news articles from different online Bangla news portals. We picked sentences and tokenized them for tagging purpose. There are many standard schemes like IOB, IOBES, BIO for tagging Named entity. We have selected BIO scheme for our tagging system as a recent study found that it performs better than other tagging schemes [24]. In this scheme, every entity token starts with a B-Tag, and if the entity consisting of more than one tokens, the followings are tagged with I-Tag. This dataset contains four types of entities as the CoNLL-2003 Shared Task [25]. These are person(PER), location(LOC), organization(ORG) and miscellaneous(MISC). Tokens do not belong to these four entity types are considered as other(O). We have followed the tagging guidelines as a study told about what to annotate [26]. We have used a crowdsourcing platform[1] to tag our dataset. We found that this dataset is not the gold standard. The dataset contains 10290 sentences and 176029 tokens. The statistics and tag distribution of the dataset are given in table I and figure I, respectively. We have excluded the other tag in the graphical representation.

Total Sentences	10290
Total Tokens	176029
Total Tags	4

TABLE I
NER dataset statistics

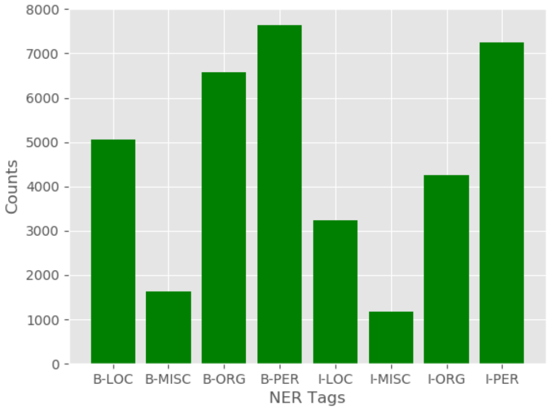

Fig. 1. NER Tag Distributions

B. POS Dataset

We have prepared our dataset for POS tagging. We have used top-level categories of the tagset proposed by the Bureau of Indian Standard (BIS) [23]. We have collected various articles from different types comprising politics, economics, entertainment, sports, lifestyle, etc. from different Bangla online newspapers. We have tokenized the sentences by following standard tokenization scheme. We have tagged 47594 tokens of 4944 sentences. We have included the corpus statistics and the POS tag distribution in table II and figure II We have found the most frequent tag in our corpus is the tag ’Verb’ and it occurred in 12251 tokens.

[1] crowd.pipilika.com

Total Sentence	4944
Total Tokens	47615
Total Tags	12

TABLE II
POS DATASET STATISTICS

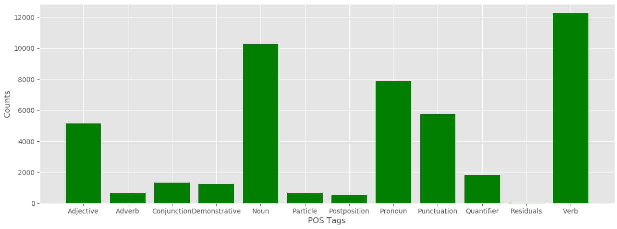

Fig. 2. POS Tag Distributions

IV. Neural Network Architecture

We have used traditional deep neural networks CNN, LSTM, BLSTM in our experiments. Brief descriptions of the models that we have used are given below.

A. CNN

In recent studies found that the CNN is not only useful for computer vision but also very much effective in extracting morphological information a word [27], which can be used for propagating character representations of words into neural networks.

B. LSTM

RNNs are very efficient in capturing long-distance dependencies, and one of its variant LSTM solved its vanishing gradient problem [28] - [30]. LSTM cells carry past information through themselves. Each cell has different gates to interact with the data passing through it. LSTMs cell can update, remove the portion of data using the gates. A schematic design of a basic LSTM cell is given in figure III.

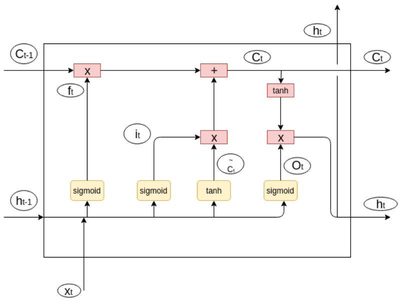

Fig. 3. Basic LSTM cell

The equations used by an LSTM cell to perform operations are given below.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{2}$$

$$C_t^{\sim} = tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * C_t^{\sim} \tag{4}$$

$$o_t = \sigma(W_o.[h_{t1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

Here σ symbolizes element-wise sigmoid operation. And . shows the element-wise product. C_t and x_t stand for the context vector and the input vector at time t

C. BLSTM

Generally, LSTM cells only have information about the past context. Its hidden states do not have any information about the future context. Dyer et al. proposed an elegant solution which uses the sequence forwards and backward to two different states and concatenates them to form the final output [31] which is Bi-directional LSTM(BLSTM)

D. CRF

Contemplation of the correlation between neighborhood labels and jointly decode the best chain of labels for a stated input sentence is useful for general structured prediction or sequence labeling tasks [32]. A coherent instance is- in the case of POS tagging, an adjective presumably followed by a noun than a verb, as well as I-ORG, refrain following I-PER on NER with standard BIO annotation.

V. Models, Training & Hyperparameters

We have tried five models on both tasks. The first three of them were CRF on top of BLSTM. The first one was without pre-trained embedding and character embedding. The second model was without character embedding. The third one included both character embedding and pre-trained embedding. The fourth model we have tried CNN on top of BLSTM with character embedding, and pre-trained embedding included. The final one we have used was CNN on top character embedding and BLSTM on top of the concatenation of word and character embedding after applying CNN on it. The architecture of our BLSTM-CNN-CRF model is given in figure IV.

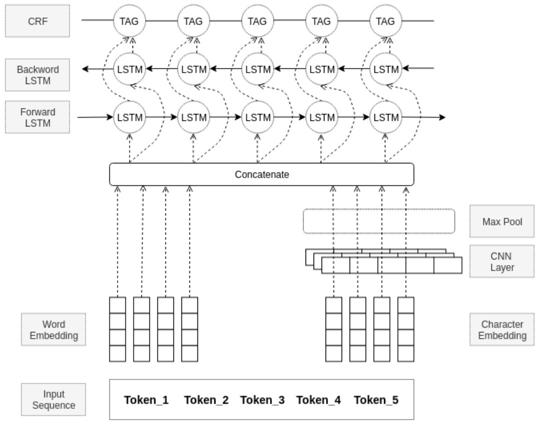

Fig. 4. BLSTM-CNN-CRF Model Architecture

We have used keras[2] for all of our training and testing. Different hyperparameter tunings were evaluated. We have tried LSTM with units 100, 75, 50 with different recurrent dropout. We have tried different kernel size for CNN and found that kernel size 5 did better. We have also tried different optimization techniques like Nadam [33], Adam [34], RMSProp [35] and found that Adam performed faster and better. As the dropout technique heavily used to reduce overfitting in deep neural networks [36], we have used it with different rate and found that dropout rate 0.5 performed best. This work [37] developed the pre-trained word embedding model we have used. We have used scikit-learn[3] train-test split API to split the dataset into train and test set and the ratio was 80 to 20. Important hyperparameters are given in the table III.

params	best performance
LSTM units	100
Char embedding size	30
CNN kernel, filters	5, 30
Optimizer	adam
Dropout rate	0.5

TABLE III
Parameters and Hyperparameters

VI. Results & Analysis

A. NER

We have used the strict f1 score as a metric for evaluation where predicted tokens of an entity must need to exact same of the true label. Partial matching of an entity is considered as a wrong prediction. We have used seqeval, an implementation of Standard NER evaluation[4]. We have found the best result for BLSTM-CNN-CRF model with a strict f1 score of .6284. We have also found that our model performs better on recognizing the most frequent entity the person entity and performs worst on identifying the least occurred tag in the dataset. The performances of the models and the result achieved by the models we have used described in table IV and table V respectively.

model	f1
BLSTM-CRF[without ce, without pwe]	47.23
BLSTM-CRF [without ce]	59.22
BLSTM-CRF	62.20
BLSTM-CNN	60.40
BLSTM-CNN-CRF	**62.84**

TABLE IV
strict f1 scores obtained by the models on NER dataset

tag	precision	recall	f1	support
LOC	0.5860	0.5402	0.5621	896
PER	0.8067	0.7759	0.7910	1178
ORG	0.5890	0.5788	0.5838	1161
MISC	0.3898	0.1631	0.2300	282
micro avg	0.6576	0.6016	0.6284	3517
macro avg	0.6452	0.6016	0.6193	3517

TABLE V
BLSTM-CNN-CRF's classification report on NER dataset

B. POS

Experimenting on the POS dataset, we have found that, like before our BLSTM-CNN-CRF based model best with an accuracy of 93.50%. As expected, it correctly detected all the punctuations of the test case. The model performances and the precision, recall, and f1 measures found from BLSTM-CNN-CRF model are given in tableVI and table VII respectively.

model	accuracy
BLSTM-CRF [without ce, without pwe]	90.04
BLSTM-CRF [without ce]	91.70
BLSTM-CRF	92.29
BLSTM-CNN	92.50
BLSTM-CNN-CRF	**93.86**

TABLE VI
Accuracies obtained by the models on POS dataset

tag	precision	recall	f1	support
Noun	0.6491	0.6677	0.6583	1252
Postposition	0.7000	0.5957	0.6437	94
Adjective	0.5896	0.5972	0.5934	854
Adverb	0.7573	0.5735	0.6527	136
Punctuation	1.0000	0.9991	0.9996	1150
Verb	0.7426	0.7233	0.7328	1388
Pronoun	0.6729	0.6232	0.6471	1205
Quantifier	0.8799	0.9071	0.8933	323
Conjunction	0.9048	0.8736	0.8889	261
Demonstrative	0.8091	0.7807	0.7946	228
Particle	0.9280	0.8056	0.8625	144
Residuals	1.0000	0.7500	0.8571	8
micro avg	0.7556	0.7390	0.7472	7043
macro avg	0.7558	0.7390	0.7468	7043

TABLE VII
BLSTM-CNN-CRF's classification report on POS dataset

VII. Conclusion

In this work, we have experimented different deep neural networks with varying parameters for solving two similar tasks POS tagging and NER in Bangla language. We have come up with BLSTM-CNN-CRF based truly end to end solution for solving these two sequence labeling tasks. For this, we have collected news articles from different categories of different Bangla online news portals and prepared two datasets using standard tags and tagging schemes. We have used BIS top POS categories for making POS tagging dataset and used BIO tagging scheme for tagging named entity. For lackings of publicly available dataset regarding these task for Bangla

[2] keras.io

[3] http://scikit-learn.org/

[4] https://github.com/chakki-works/seqeval

language, we can not compare our work with others. But we have evaluated our models on our datasets using standard train test split. Our proposed model did better in both categories. In POS tagging, our proposed model achieved an accuracy of 93.86%, and in NER, our model achieved an f1 score of .6284. This work is several directions for future research. One can try to solve the tasks without being heavily dependent on handcrafted features that require domain-specific expertise. Sentence embeddings and more complex neural networks can be applied to large datasets to achieve better performance.

Acknowledgment

This research work is partially funded by Access to Information (a2i) programme ran from the Prime Minister's office of Bangladesh.

References

[1] Manning, C.D., 2011, February. Part-of-speech tagging from 97% to 100%: is it time for some linguistics?. In International conference on intelligent text processing and computational linguistics (pp. 171-189). Springer, Berlin, Heidelberg.

[2] M. Paul Lewis, Gary F. Simons, Charles D. Fennig, "Ethnologue: Languages of the World" in Nineteenth, Dallas, Texas:SIL International, 2016.

[3] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), p.436.

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

[5] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G. and Chen, J., 2016, June. Deep speech 2: End-to-end speech recognition in english and mandarin. In International conference on machine learning (pp. 173-182).

[6] A. Ekbal and S. Bandyopadhyay, Named entity recognition using support vector machine: A language independent approach, International Journal of Electrical, Computer, and Systems Engineering, vol. 4, no. 2, pp. 155170, 2010.

[7] A. Ekbal and S. Bandyopadhyay, Bengali named entity recognition using classifier combination, in ICAPR. IEEE, 2009, pp. 259262.

[8] A. Ekbal and S. Bandyopadhyay, Named entity recognition in bengali: A multi-engine approach, Northern European Journal of Language Technology, vol. 1, no. 2, pp. 2658, 2009.

[9] A. Ekbal and S. Bandyopadhyay, A web-based bengali news corpus for named entity recognition, Language Resources and Evaluation, vol. 42, no. 2, pp. 173182, 2008.

[10] A. Ekbal and S. Bandyopadhyay, Development of bengali named entity tagged corpus and its use in ner systems, in Proc. of the 6th Workshop on Asian Language Resources, 2008.

[11] A. Ekbal and S. Bandyopadhyay, Bengali named entity recognition using support vector machine, in Proc. of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages, 2008.

[12] A. Ekbal, R. Haque, and S. Bandyopadhyay, Named entity recognition in bengali: A conditional random field approach, in Proc. of the 3rd Joint Conference on NLP, 2008.

[13] A. Ekbal and S. Bandyopadhyay, A hidden markov model based named entity recognition system: Bengali and hindi as case studies, in International Conference on PRML. Springer, 2007, pp. 545552.

[14] B. B. Chaudhuri and S. Bhattacharya, An experiment on automatic detection of named entities in bangla, in Proc. of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages, 2008.

[15] K. S. Hasan, V. Ng et al., Learning-based named entity recognition for morphologically-rich, resource-scarce languages, in Proc. of the 12th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2009, pp. 354362.

[16] Ekbal, A., Haque, R. and Bandyopadhyay, S., 2007, December. Bengali part of speech tagging using conditional random field. In Proceedings of Seventh International Symposium on Natural Language Processing (SNLP2007) (pp. 131-136).

[17] Ekbal, A., Haque, R. and Bandyopadhyay, S., 2008. Maximum entropy based bengali part of speech tagging. A. Gelbukh (Ed.), Advances in Natural Language Processing and Applications, Research in Computing Science (RCS) Journal, 33, pp.67-78.

[18] Ekbal, A. and Bandyopadhyay, S., 2008, December. Part of speech tagging in bengali using support vector machine. In 2008 International Conference on Information Technology (pp. 106-111). IEEE.

[19] Dasgupta, S. and Ng, V., 2007, June. Unsupervised part-of-speech acquisition for resource-scarce languages. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL) (pp. 218-227).

[20] Mukherjee, S. and Mandal, S.K.D., 2013, December. Bengali parts-of-speech tagging using global linear model. In 2013 Annual IEEE India Conference (INDICON) (pp. 1-4). IEEE.

[21] Alam, F., Chowdhury, S.A. and Noori, S.R.H., 2016, December. Bidirectional lstmscrfs networks for bangla pos tagging. In 2016 19th International Conference on Computer and Information Technology (ICCIT) (pp. 377-382). IEEE.

[22] Bali, M.K. and Biswas, P., 2010. Indian language part-of-speech tagset: Bengali ldc2010t16. In Philadelphia: Linguistic Data Consortium, Tech. Rep..

[23] Dash, N.S., 2013. Part-of-Speech (POS) Tagging in Bengali Written Text Corpus. International Journal on Linguistics and Language Technology, 1(1), pp.53-96.

[24] Reimers, N. and Gurevych, I., 2017. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. arXiv preprint arXiv:1707.06799.

[25] Sang, E.F. and De Meulder, F., 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050.

[26] Fort, K., Ehrmann, M. and Nazarenko, A., 2009, August. Towards a methodology for named entities annotation. In Proceedings of the Third Linguistic Annotation Workshop (pp. 142-145). Association for Computational Linguistics.

[27] Yin, W., Kann, K., Yu, M. and Schtze, H., 2017. Comparative study of CNN and RNN for natural language processing. arXiv preprint arXiv:1702.01923.

[28] Bengio, Y., Simard, P. and Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2), pp.157-166.

[29] Gers, Felix A., Jrgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM." (1999): 850-855.

[30] Hochreiter, Sepp, and Jrgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[31] Dyer, C., Ballesteros, M., Ling, W., Matthews, A. and Smith, N.A., 2015. Transition-based dependency parsing with stack long short-term memory. arXiv preprint arXiv:1505.08075.

[32] Lafferty, J., McCallum, A. and Pereira, F.C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[33] Dozat, T., 2016. Incorporating nesterov momentum into adam.

[34] Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[35] Tieleman, T. and Hinton, G., 2012. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. University of Toronto, Technical Report.

[36] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), pp.1929-1958.

[37] Ahmad, A. and Amin, M.R., 2016, December. Bengali word embeddings and it's application in solving document classification problem. In 2016 19th International Conference on Computer and Information Technology (ICCIT) (pp. 425-430). IEEE.