A NOVEL APPROACH FOR NAMED ENTITY RECOGNITION ON HINDI LANGUAGE USING RESIDUAL BILSTM NETWORK

Rita Shelke[1] and Devendrasingh Thakore[2]

[1]Research Scholar, Pune, India [2]Head, Department of Computer Engineering, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India

ABSTRACT

Many Natural Language Processing (NLP) applications involve Named Entity Recognition (NER) as an important task, where it leads to improve the overall performance of NLP applications. In this paper the Deep learning techniques are used to perform NER task on Hindi text data as it found that as compared to English NER, Hindi language NER is not sufficiently done. This is a barrier for resource-scarce languages as many resources are not readily available. Many researchers use various techniques such as rule based, machine learning based and hybrid approaches to solve this problem. Deep learning based algorithms are being developed in large scale as an innovative approach now a days for the advanced NER models which will give the best results out of it. In this paper we devise a Novel architecture based on residual network architecture for preferably Bidirectional Long Short Term Memory (BiLSTM) with fasttext word embedding layers. For this purpose we use pre-trained word embedding to represent the words in the corpus where the NER tags of the words are defined as the used annotated corpora. BiLSTM Development of an NER system for Indian languages is a comparatively difficult task. In this paper, we have done the various experiments to compare the results of NER with normal embedding and fasttext embedding layers to analyse the performance of word embedding with different batch sizes to train the deep learning models. Here we present a state-of-the-art results with said approach F1 Score measures.

KEYWORDS

Natural Language Processing, Named Entity Recognition, Residual Network, Machine Translation

1. INTRODUCTION

Named Entity Recognition (NER) was first introduced in 1995 in (MUC-6) Message Understanding Conference-6 (MUC-6, 1995). [8] Where it is stated as it is consisting of three sub tasks, and these tasks are namely, i) entity names, ii) temporal expressions and iii) number expressions. where the terms to be annotated are as unique identifiers like (a) entity names like the names of organizations, the names of persons or the names of locations etc. (b) temporal expressions like times and dates, and (c) number expressions or quantities like monetary values, percentages. Hence NER is one of the key tasks in the field of information extraction and Natural Language Processing (NLP). English language can boast of a rich NER literature, however, the same cannot be said to be true for Hindi language. There have been periodical attempts, as there is big scope to explore in the Hindi language domain, while considering especially the use of deep learning models have made their way to resolve several language processing problems. Due to Lack of availability of ready tools, rich morphology nature of Hindi language and more precisely the scarcity of annotated corpus data makes it i) difficult to reuse existing deep learning

DOI: 10.5121/ijnlc.2020.9201

architectures which are used for English language are more challenging and (b) allows exploring novel and advanced approaches being used for NER task.

Based on the success of using machine learning architectures for NER task, for resource rich languages like English, in this paper we follow a simple and effective approach of refining previously proven successful deep neural network models for Hindi language. The idea behind this is to use fasttext embedding structure with residual deep neural network architecture which is novel in nature and which is easy to optimise the model parameters in low-resource scenario. As we design increasingly deeper networks it becomes imperative to understand how adding layers can increase the complexity and expressiveness of the network. Even it is more important that the ability to design networks where adding layers makes networks strictly more expressive rather than just different. The architecture geared towards low resource data and less resources in terms of computing time and power but also shows an improvement over the existing models for the Hindi NER task. We show experimentally that there is an improvement in Hindi NER performance over the base BiLSTM model by adding residual connections, which is the main contribution of this paper. Deep residual networks were shown to be able to scale up to thousands of layers and still have improving performance. [12] We believe that these kinds of modifications or integration of different network models help improve Hindi NER performance especially in low-resource conditions.

2. RELATED WORK

Development of an NER system for Indian languages is a comparatively difficult task.

Hindi and many other Indian languages provide some inherent difficulties in many NLP related tasks. Consequently, not much work has been done on NER for Indian languages like Hindi. Hindi is the third most spoken language of the world and still no accurate Hindi NER system exists. As some features like capitalization are not available in Hindi and due to lack of a large labelled dataset [11] and of standardization and spelling variations, an English NER system cannot be used directly for Hindi.

Furthermore, the structure of the language contain many complexities like free word ordering (which affect ngram-based approaches significantly) and its inflectional nature (affecting hand-engineered approaches significantly). Also, in Indian languages there are many word constructions that can be classified as Named Entities (Derivational/Inflectional constructions) and these constraints on these constructions vary from language to language hence carefully crafted rules need to be made for each language which is a very time consuming and expensive task. Also, the scarcity of labelled data renders many of the statistical approaches like Deep Learning unusable. This complexity in the task is a significant challenge to solve. However, Shah et. al. have demonstrated promising results by utilizing BiLSTM networks to solve the NER problem [5], our work builds upon theirs and adds residual connections to the network.

There is a need to develop an accurate Hindi NER system for better presence of Hindi on the Internet. It is necessary to understand Hindi language structure and learn new features for building better Hindi NER systems.

3. MATERIAL AND METHOD

3.1. Word Embeddings

Word embeddings are an efficient way to represent words - i.e. words with same meanings are represented in the same way which is useful for various NLP tasks. As the quality of word embeddings depends upon the quality of input data, hence representing the data in the form of words is the essential task and now a days embeddings of words into low dimensional space is mostly suggested. Recently word embeddings like Distributed word representations have contribution to competitive performance in language modeling and with various NLP tasks. There are many neural network embedding approaches where as the skip-gram model of has achieved significant results in many NLP tasks, where it includes sentence completion, analogy and sentiment analysis etc. Word2vec is a statistical method for learning word embeddings from a large text corpus. It outputs a high-dimensional vector space, where each word from the corpus is assigned a vector and words with common contexts are placed proximally close in the vector space. [1]

We have chosen Fasttext, a pre-trained word embedding developed and open-sourced by Facebook [2] for our task. As already fasttext approach for English language NER has given results which are comparatively better than regular methods used for Named entity recognition. But in regional language like Hindi it is found that due to the unavailability of large corpus of data the experiments are done with regular Deep learning algorithm with traditional approach. Here, we use novel architecture to analyse the performance of NER w.r.t. BiLSTM neural network. It provides word embeddings for Hindi (and 157 other languages) and is based on the CBOW (Continuous Bag-of-Words) model. The CBOW model learns by predicting the current word based on its context, and it was trained on Common Crawl and Wikipedia. [3]

3.2. Dataset

We perform the task of labelling the named entities on the dataset, available at [4], released during ICJNLP 2008 as part of the workshop on NER for South and South East Asian Languages, consisting of 19822 annotated sentences, 490368 total tokens among which 34193 are unique tokens, and 12 categories of entities and one negative entity class other. The 12 categories are given in Table 1

Table 1. Categories in the dataset

Tag	**Category**
NEP	Person
ED	Designation
NEO	Organization
NEA	Abbreviation
NEB	Brand
NETP	Title-Person

NETO	Title-Object
NEL	Location
NETI	Time
NEN	Number
NEM	Measure
NETE	Term

“इस रोगी की आयु ४२ (NEM) वर्ष थी और उनको पिछले छह (NEM) महीने से इस बात की शिकायत थी” is a sample sentence in the dataset.

We faced a number of issues while working with the IJCNLP dataset.

- More than 80% of the words do not have tags.
- Many sentences contain English language words.
- It is not clear if words without tags have not been tagged or if they belong to {\tt other} category
- More than 5,000 sentences in the dataset are with no tags

3.3. Pre-Processing Steps

The dataset was in Shakti Standard Format (SSF) but could not directly be fed into a model, so it needed parsing, which was carried out with handwritten Regex parsers in Python.
Steps involved in pre-processing the data

- Parsing SSF
- Removing sentences with no tags, after which 7966 sentences remained.
- Mapping all words to numbers which would then be mapped to their respective embeddings with each embedding of dimension 300 for Fasttext
- Padding sentences with "0" and truncating sentences so that all sentences are of same length, i.e. 30
- The dataset was split in a 70:15:15 ratio for training, testing and validation sets respectively.

3.4. Mathematical Algorithms Used

1) Softmax Activation Function: For activation, our model uses the Softmax function. It is a type of activation function used in Neural Networks. It is used to compute probability distribution from a vector of numbers. It produces an output between 0 to 1, and the sum of probabilities are equal to 1. The Softmax activation function is computed using the following relationship.

$$f(x_i) = \frac{exp\ (x_i)}{\sum_j exp\,(x_j)}$$

The Softmax function is used in multi-class models where it returns probabilities of each class, with the target class having the highest probability.

In most cases, the Softmax function shows up in the output layers of deep learning architectures, even in ours.

2) Recurrent Dropout: Recurrent dropout is an method that can preserve memory in an LSTM while still generating different dropout masks for each input sample. Recurrent dropout works by selectively applying dropout to that part of the Recurrent Neural Network which is updating the hidden state, as opposed to the state itself. Thus, a dropped element does not contribute to the network's memory and does not erase the hidden state. For LSTM, the equation is same as vanilla LSTM, except that the equation for Ct changes.

3.5. Proposed Approach

Previous works have used Bi-LSTM networks for Hindi NER, but our approach builds on it and adds residual connections to the model. The input is in the form of batches of Hindi sentences in which there is a mapping of numbers to words which is then passed to the embedding (fasttext) layer wherein each number is mapped to a specific vector i.e., each word is mapped to a learned vector in fasttext. To get a deeper representation of the words, we have used a residual connection architecture of two layers which was obtained by adding the output of the first layer to the stacked output of the second layer to get a deeper representation. This residual connection allows the model to get a deeper understanding of the context of the words and improves the performance by increasing the precision score from 78% to 81.9% as compared to the work done by Shah et. al. [5] In order to counter over fitting, we have added a dropout layer after the residual connection and used recurrent dropout in the recurrent layers. At the end of the model, we have used a time distributed dense layer so as to map each word representation in the sentence to a dense layer and from there to an output tag probability for each word.

A plot of the model can be seen in Figure 1.

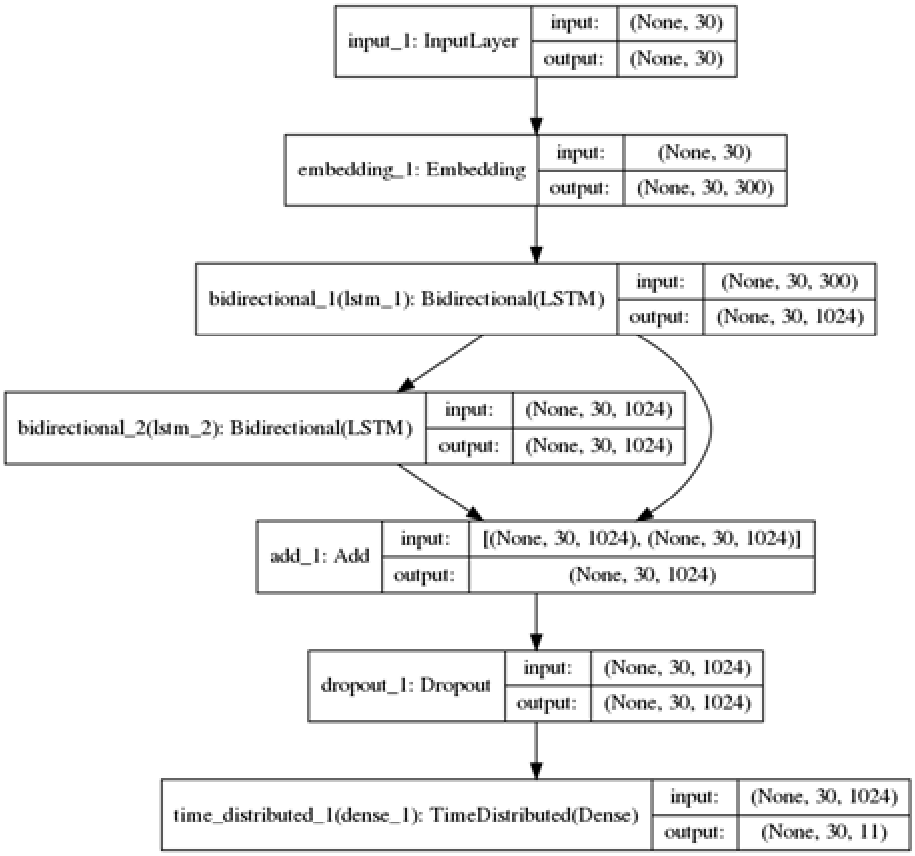

Figure 1. Layers of the Deep Learning Model

4. EXPERIMENTAL RESULTS

Residual Connection

4.1. Hardware Setup

The models were trained on an MSI laptop having specifications given in Table 2. Due to the heavy word embedding dimensions, it is advisable to carry out the training process on GPUs only.

Table 2. Hardware Setup

Type	Details
Memory	7.6 GB
Processor	Intel Core i5-9300H

CPU	@ 2.4 Ghz * 8 (cores)
Software	Keras and Tensorflow running on GPU with CUDA 10.2
GPU	GeForce GTX 1050 Ti/PCle/SSE2

4.2 . Results Obtained and Their Analysis

The model was trained on 12,464,023 parameters with varying batch sizes and was subject to testing on each. The best results were obtained with batch size 32 and at 5 epochs. The metrics have been calculated on a single fit. Cross validation was not carried out because the dataset is large enough. The results are tabulated and shown in Table 3. The precision was found to be higher by 3.9% than that of previous work done on BiLSTMs for NER. [5]

Table 3. Results and Analysis

Metric	Values
F1-score	69.5%
Accuracy-score	96.8%
Precision-score	81.9%
Recall-score	60.4%

5. Conclusion

Most of the NLP applications in Computer Science have their first step rooted in Named Entity Recognition. However, there is a lack of collated information on NER methods used for processing Hindi.This is one of the first attempts at applying residual connections to BiLSTM networks for NER task.It has been shown that rule-based approaches outperform others if expert linguists are available, but with advances in machine learning and deep learning models, this situation is soon to change, for a large set of languages.

References

[1] Mikolov, Tomas, et al. “Efficient Estimation of Word Representa-tions in Vector Space.” ArXiv:1301.3781 [Cs], Sept. 2013. arXiv.org,http://arxiv.org/abs/1301.3781

[2] Bojanowski, Piotr, et al. “Enriching Word Vectors with Subword Information.” ArXiv:1607.04606 [Cs], June 2017. arXiv.org, http://arxiv.org/abs/1607.04606.

[3] Grave, Edouard, et al. “Learning Word Vectors for 157 Languages.” ArXiv:1802.06893 [Cs], Mar. 2018. arXiv.org, http://arxiv.org/abs/1802.0689 3.

[4] IJCNLP-08 Workshop on NER for South and South East Asian Languages. http://ltrc.iiit.ac.in/ner-ssea-08/. Accessed 29 Feb. 2020.

[5] Shah, Bansi, and Sunil Kumar Kopparapu. "A Deep Learning Approach for Hindi Named Entity Recognition." ArXiv:1911.01421 [Cs], Nov. 2019. arXiv.org, http://arxiv.org/abs/1911.01421.

[6] Xie, Jiateng, et al. "Neural Cross-Lingual Named Entity Recognition with Minimal Resources." ArXiv:1808.09861 [Cs], Sept. 2018. arXiv.org, http://arxiv.org/abs/1808.09861.

[7] P, Praveen, and Ravi Kiran V. "Hybrid Named Entity Recognition System for South and South East Asian Languages." Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages, 2008. ACLWeb, https://www.aclweb.org/anthology/I08-5012.

[8] MUC-6. 1995. Named Entity Task Definition. 6th Message Understanding Conference.

[9] Isozaki, Hideki, and Hideto Kazawa. "Efficient Support Vector Classifiers for Named Entity Recognition." Proceedings of the 19th International Conference on Computational Linguistics -, vol. 1, Association for Computational Linguistics, 2002, pp. 1–7. DOI.org (Crossref), doi:10.3115/1072228.1072282.

[10] Fernandes, Ivo, et al. "Applying Deep Neural Networks to Named Entity Recognition in Portuguese Texts." 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS), IEEE, 2018, pp. 284–89. DOI.org (Crossref), doi:10.1109/SNAMS.2018.8554782.

[11] Athavale, Vinayak, et al. "Towards Deep Learning in Hindi NER: An Approach to Tackle the Labelled Data Sparsity." Proceedings of the 13th International Conference on Natural Language Processing, NLP Association of India, 2016, pp. 154–160. ACLWeb, https://www.aclweb.org/anthology/W16-6320.

[12] Zagoruyko, Sergey, and Nikos Komodakis. "Wide Residual Networks." Procedings of the British Machine Vision Conference 2016, British Machine Vision Association, 2016, pp. 87.1-87.12. DOI.org (Crossref), doi:10.5244/C.30.87.