International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Development of Part of Speech Tagger Using Deep Learning

Aarti Singh, Charu Verma, Swagata Seal, Varsha Singh

Abstract: *Part of speech tagging is the initial step in development of NLP (natural language processing) application. POS Tagging is sequence labelling task in which we assign Part-of-speech to every word (Wi) which is sequence in sentence and tag (Ti) to corresponding word as label such as (Wi/Ti…. Wn/Tn). In this research project part of speech tagging is perform on Hindi. Hindi is the fourth most popular language and spoken by approximately 4billion people across the globe. Hindi is free word-order language and morphologically rich language due to this applying Part of Speech tagging is very challenging task. In this paper we have shown the development of POS tagging using neural approach.*

Keywords **:** *POS Tagging, LSTM, RNN, Hindi*

I. INTRODUCTION

Part of speech define how the word is used in a sentence. There are eight part of speech Nouns (naming word), Pronouns (use in place of noun), Adjectives (describing word), Verbs (action word), Adverbs (describes a verb), Prepositions (shows relationships), Conjunctions (joining word) and Interjections (Expressive word). Part of speech define the categories of word.

There are two categories of Part of speech tagging: Open class and close class. These are Open class, which has words from POS categories like Noun, Verb, Adjectives and Adverbs; and Closed class, which has words for rest of the POS categories.

Part of speech tagging is also called lexical categories, word classes, morphological classes, lexical tags, grammatical tagging. Part of speech tagging is a process that attach each word of sentence with a suitable tag from a given set tags. The set of tags is called tag-set. Here Tagging mean assign single part-of-speech to each word (punctuation marker) in a corpus. Standard Tag-set for English is Penn Treebank and for hind language is ILPOS tag (Indian Language part of speech tagging).

POS Tagging is sequence labelling task in which we assign Part-of-speech to every word (Wi) which is sequence in sentence and tag (Ti) to corresponding word as label such as (Wi/Ti…. Wn/Tn). To assign a POS tag is not an easy task. While tagging a sentence many ambiguities occur. For example.

In English: I $bank_1$ on the $bank_2$ on the river $bank_3$ for my transaction.

Here Bank1 is verb and other two bank is noun.

In Hindi: “खाना”: can be noun (food) or verb (to eat)

राम अच्छा गाता है (है is the VAUX (Auxiliary verb))

राम अच्छा लड़का है (हैVCOP (Copula verb))

In the above two sentences both the है have different tagging.

When the word display ambiguity across POS category is known as inter-POS ambiguity. This word has multiple entry in the lexicon (one for each category). when we perform stemming on the word then after stemming the word will assign all possible POS tag according to the number of entries in the lexicon. For example: “I get bank to the bank seat to give rest to my back” in this example bank appear in three position which different-different POS categories.

So, complexity occur when we it deals with the morphologically rich language and free word order language such as hind. In which all the order of the words in a clause are possible. This is the challenge where tagging of POS tag through stochastic tagging become difficult. Intra-POS ambiguity is occurred when a word has one POS with different-different feature value. Example: लड़केin hind is noun but can be featured in two way. 1. POS: Noun, Number: singular, Case: Oblique. मैंने लड़के को आम दिया। 2.POS: Noun, Number: plural, Case: Direct लड़के आम खाते हे| It is difficult to assign appropriate tag on the base of structure of word and the context used.

To define whether the tag of the word is adjective or adverb is more complex than to define noun and verb. For example: परcan occur as conjunction, post-position or noun and falls in ambiguity schemes ‘conjunction-noun-postposition’. We grouped all the Ambiguity words into set according to the Ambiguity Scheme that are possible in hind, eg, Adjective-Noun, Adjective-Adverb, Noun-Verb etc.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Aarti Singh*, Department of Computer Science, Banasthali Vidyapith, India. Email: vermacharu284@gmail.com

Charu Verma, Department of Computer Science, Banasthali Vidyapith, India. Email: say2aru19@gmail.com

Swagata Seal, Department of Computer Science, Banasthali Vidyapith, India. Email: swagata.sita@gmail.com

Varsha Singh, Department of Computer Science, Banasthali Vidyapith, India. Email: varshasingh773@gmail.com

II. LITERATURE REVIEW

In this section we will discuss work done on POS tagging in Indian language using different-different approach. Joshi [1] defined POS tagging for Hindi language using HMM base tagger with 24tagset and attend accuracy of 92.13%. Modi [2] This tagger accepts data in Devanagari Hindi use Rule based approach which includes grammatical rules and regular expression,100% correctness for split and tokenizer,91.84% for POS tagging. This system achieved 91.84% of average precision and 85.45% of average accuracy. Ekbal [3] statistical maximum entropy model.13 is used for Bengali Part of Speech Tagging, ME system has been compared with HMM based POS tagger in which trigram model is considered. Performance of ME POS tagger is better than HMM based POS tagger and also worked with [7] Conditional Random Field features that used *Context word feature, *Word suffix, *POS information *Name entity information, *Length of word, *Lexicon feature and perform work with two different data set with the following accuracy of 92.1% and 90.3% etc. Dandapat [4] developed a hybrid tagging model, the training model is based on partially supervised learning, for supervised and unsupervised learning HMM is used. This POS tagger dealt with Chalitabhasha of Bengali with 40 different tags are used.

Singh [5] POS tagger for morphologically rich language Hindi using Decision tree-based learning algorithm CN2(Clark and Niblett, 1989). The accuracy obtained by simple lexicon lookup, based approach is 61.19%. If multiple tags for a word is considered as an error then the accuracy is 73.62%. Final accuracy was gained by the learning-based tagger after 4-fold cross validation, and the result is 93.45% which is quite satisfactory. Dandapat [6] worked on POST for Bengali with HMM use combination of supervise (HMM-S) and unsupervised (HMM-SS) model. These models are experimented with (HMM-S+IMA and HMM-SS+IMA) show complete morphological restriction and (HMM-S-CMA and HMM-SS-CMA) show complete morphological restriction. Singh [8] POS tagger for Marathi language using Trigram model Corpus consists of 2000 sentences i.e. 48,635 words Accuracy gained is 91.63%.

Patil [9] use rule base POS tagger for Marathi language, corpus consists of 576 unique words and are manually tagged by 9 tag sets and for evaluation of the system three different tag sets are formed. Joshi et al [9] developed a mechanism to write in Hindi using English. They used statistical machine learning to predict a word when some of the initial characters are typed. Using this Joshi et al. [10] also developed an Example Based Machine Translation System. Joshi et al. [11] also evaluated the system developed. They also compared the performance of this system with other popularly available MT engines. Gupta et al. [12] developed a rule-based stemmer for Urdu. They developed several rules to implement this stemmer. They further used this stemmer in evaluation of some English-Urdu MT systems [13]. Singh et al. [14] developed a POS tagger for Marathi using Statistical Machine Learning. Bhalla et al. [15] developed a procedure of transliteration of name entities from English to Punjabi. Joshi et al [16] evaluated several open domain MT engines. Gupta et al. [17] did the same for English-Urdu MT engines. Singh et al. [18] developed a POS tagger for Marathi using supervised learning. Joshi et al. [19] further developed a technique to using machine learning in evaluating MT engines. Tyagi et al. [20] [21] developed an approach of translating complex English sentences by first simplifying them and then translating into Hindi. Yogi et al. [22] developed an approach to identify candidate translation which are good for post editing. Gupta et al. [23] further extended their stemmer by adding derivational rules to the inflectional stemmer. Asopa et al. [24] developed mechanism for chunking Hindi sentences using a rule-based approach. Gupta et al. [25] developed a rule based lemmatizer for Urdu which was an extension to their stemmer. Kumar et al. [26] developed several machine learning based classifiers for identifying different senses to a word in Hindi. Joshi et al. [27] developed a mechanism to estimate the quality of English-Hindi MT engines. Chopra et al. [28] [29] developed a name entity recognition and tagging tool for Hindi using several machine learning approaches. Gupta et al. [30] developed a POS tagger for Urdu using machine learning approach. Mathur et al. [31] developed an ontology matching evaluation using tool which used the MT engine developed by Joshi et al. Chopra et al. [32] developed a mechanism for rewriting English sentence and then translating them into Hindi. This significantly improved the performance of their MT engine. Joshi et al. [33] investigated some approaches to classifying documents and further suggested an approach for effective classification of text documents.

Singh et al. [34] developed an approach to automatically generate transfer grammar rules. This approach significantly improved the development process of their transfer-based MT engine. Singh et al. [35] developed an approach for text processing of Hindi documents using deep neural networks. They further developed this approach to mine textual data from web documents [36]. Singh et al. [37] developed a translation memory tool which worked as a sub-system in their transfer-based MT system. This further improved the accuracy of their system. Gupta et al. [38] further showed how fuzzy logic can be used in developing NLP applications. Gupta et al. [39] used several NLP tools in preprocessing the tweets that they extracted from web. They found that this approach improves the accuracy of their machine learning model which classifies the tweets. Gupta et al. [40] developed an approach which helped in identification and classification of multiword expressions from Urdu documents. Nathani et al. [41] developed a rule based inflectional stemmer for Sindhi which was written in Devanagari script. Asopa et al. [42] developed a shallow parser for Hindi using conditional random fields. Gupta et al. [43] showed the use of machine learning approached in developing NLP applications. Gupta et al. [44] used fuzzy operations in analyzing sentiments of tweets on several topics. This approach showed very promising results over traditional approaches. Sharma and Joshi [45] developed a rule-based word sense disambiguation approach for Hindi. It gave an accuracy of 73%. Katyayan and Joshi [46] studied various approaches of correct identification of sarcastic phrases in English documents. Gupta and Joshi [47] showed show tweets can be classified using NLP techniques. They showed how negative sentences can be handled using NLP approaches. Shree et al. [48] showed how there is difference between Hindi and English languages what problems the current state of the art MT system face while translating text.

Ahmed et al. [49] showed how MT system can be developed by using an intermediate language which is related to both the languages. They developed a Arabic-Hindi MT system using Urdu as the intermediate language. They further performed the same study using English and found that if we have a large sized corpus then English which in unrelated to Arabic and Hindi, can be used for developed a MT system [50]. Seal and Joshi [51] developed a rule based inflectional stemmer for Assamese. This system showed very good results. Singh and Joshi [52] showed the developed of POS taggers for Hindi using different markov models. They concluded that hidden markov model-based tagger produced the best results among several markov based POS taggers. Pandey et al. [53] showed how NLP approached can help in develop a better ranking model for web documents. They used particle swam optimization and NLP approaches in improving the performance of their ranking model. Singh and Joshi [54] developed a rule-based approach for identifying anaphora in Hindi Discourses. Sinha et al. [55] developed a sentiment analyzer for Facebook post using the methods developed by Gupta et al. Sharma et al. [56] [57] used some of the markov model-based approaches used by Singh et al. to develop their association classification model. Similar approaches we used by Goyal et al. [58] [59] for their models.

Zhang [60] In this paper they propose a Novel multi-channel model to handle tokens-tag dependencies and their interaction simultaneously.Basically, they studied on (POS) and (NER) Tag Dependencies.They use the Wall Street Journal (WSJ) portion of Penn Treebank for POS tagging task and standard data set splits: section 0-18 as training data, 19-21 as development data and section 22-24 as test data.They conduct NER experiments on two corpora, a Chinese corpus,People's Daily (PD)1 and an English corpus, CoNLL. For People's Daily, the data set is divided as 96750 sentences for training, 1000 sentences for validation,and 20336 sentences for test.In general ,isolated tag LSTM (MODEL II and III) achieves better performances,compared with the shared tag LSTM(model I). Results are as follow:POS tagging accuracy on WSJ with model III(JTD) 97.59 and F-score for NER:corpus (PD),model (model III,JTD), F-score (96.79) and corpus CoNLL03 model (model III+char(JTD)) F-score(91.22). Vaswani [61] In this paper they present new state-of-the-art performance on CCG super tagging and parsing. There model outperforms existing approaches by an absolute gain of 1.5%. They use feed-forward neural network models and bidirectional LSTM (bi–LSTM) based models in this work.In Feed-Forward POS tagging results.They achieve 97.28% on the development set and 97.4% on test.Model (bi–LSTM–LM+ss–train–1 model with beam decoding) has a test accuracy of 94.5%, 1.5% for state-of-the-art.Parsing results for both (Xu et al., 2015(dev(86.25),test F1(87.04))) and bi–LSTM–LM +ss–train–1 (dev(87.75),test F1(88.32)).

Tan [62] In This paper they present a comparative study of three methods to solve the problem of Malay part-of-speech (POS) tagging. These methods are LSTM networks, weighted finite state transducer (WFST) and hidden Markov model (HMM).The Malay POS annotated text used in the experiments consists of 423,767 sentences, which were tagged using 36 POS tags. The experiment results show that LSTM networks that are trained without any explicit morphological knowledge perform nearly equally with WFST but better than HMM approach that is trained with morphological information. Error rate for following models are: POS Tagging Error Rate with WFST (Phonetisaurus) higher order 6-grams gives the best result with 18.6% tagging errors in TC1 and 11.4% in TC2.POS Tagging Error Rate with HMM (Moses MT) TC1 4-gram 16.7%,TC2 4-gram 14.6%.The best result using 256 size cells, where the error rate stands at 16.2% for TC1 and 16.6% for TC2.The results show that increasing the size of LSTM cell will reduce the POS tagging error rate.

Perez-Ortiz [63] Experiments were performed to compute the error rates when tagging text taken from the Penn Treebank corpus: Random tagging incorrect tag rate between 61.8% and 62.9%. Winner-takes-it -all error rate 20.5%, HMM function of Baum-Welch iterations 8(iterations) (45.5) error rate.

Huang [64] In this paper they use variety of neural network-based model to sequence tagging task are LSTM, Bi-LSTM, LSTM-CRF, LSTM-CRF,Bi-LSTM-CRF. Bi-LSTM-CRF model use both past and further features.All model uses generic SGD forward and backword training procedure.This all model test on Penn TreeBank (PTB) POS tagging,CoNLL 2000 chunking, and CoNLL 2003 named entity tagging.Random and senna word embedding are used in above model.For random category,CRF models outperform Conv-CRF models for all three data sets. For Senna category, CRFs outperform Conv-CRF for POS task, while underperform for chunking and NER task. LSTM-CRF models outperform CRF models for all data sets in both random and Senna categories.Comparison of tagging accuracy of different models for POS.BI-LSTM-CRF accuracy(97.43) BI-LSTM-CRF (Senna) accuracy(97.55). The bi-directional LSTM CRF model, we consistently obtained better tagging accuracy than a single CRF model with identical feature sets.

Zheng [65] In this paper they explore the feasibility of performing Chinese word segmentation (CWS) and POS tagging by deep learning.They describe a perceptron-style algorithm for training the neural network it speeds up the training of the neural network and easy to implemented.The Chinese word segmentation and POS tagging can effectively perform by the deep learning.There model achieve close stat0065-of-art performance by transferring the unsupervised internal representation of Chinese characters into supervised models.The network is trained by maximizing a likelihood over all the sentences in the training set R with respect to Θ

III. PROPOSED METHODOLOGY

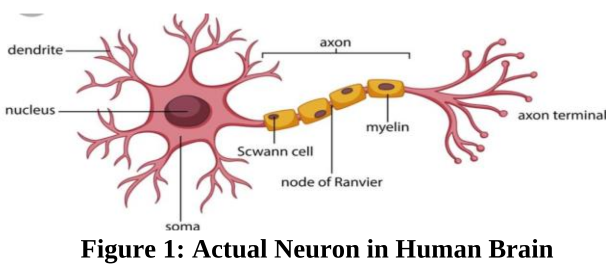

Figure 1: Actual Neuron in Human Brain

Neural network is a mathematical model inspired by the human brain. This network has input layer, an output layer and at least one hidden layer in between. But in the deep neural network we have multiple hidden layer.

As in our brain there are many numbers of neuron are present to perform functioning. Figure 1 shows the diagram of actual neuron of our brain this neuron is simulate in our artificial neural network. Same way as input neuron, hidden layer (hidden neuron) and output in neural network. Figure 2 represent the artificial neural network.

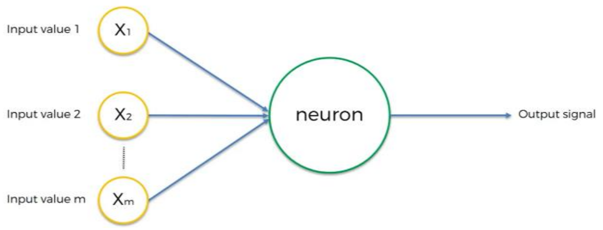

Figure 2: Architecture of an Artificial Neuron

The yellow circle is input neuron passed the signal to neuron through synapse. This input is standardized between 0 and 1. The output we get can ne categorical, binary and continues. The neuron has two steps in first the input neuron is multiplied with the assigned weight. In neural network we take input neuron every input neuron is assigned a weight this weight is then multiply with input neuron and then output is weighted sum of all input neuron. Weights are basically made the neural network to learn by considering which signal is important or which signal is not important. Figure 3 shows the process of weight calculation.

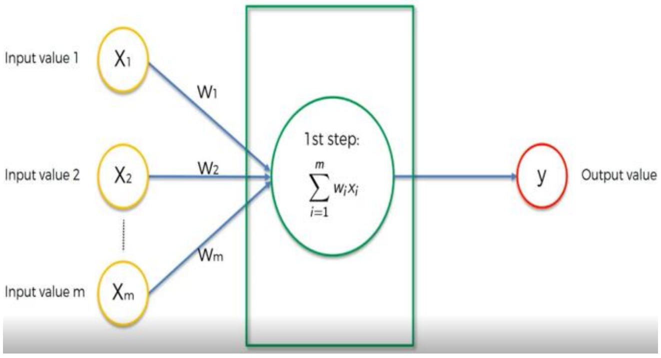

Figure 3: Weight Calculation in Artificial Neuron

In the second step we have activation function. The activation function introduces non-linearity into the output of a neuron. So that neural network capable to learn and perfume more complex tasks. It defines whether the neuron should be active or not by calculating weighted sum and adding bias with it. There are number of activation function such as threshold, sigmoid, rectifier, hyperbolic tangent activation function etc. This is shown in figure 4.

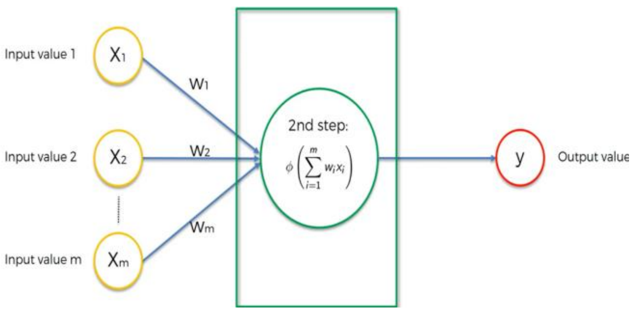

Figure 4: Activation Function in Artificial Neuron

There are different types of neural networks like Multiple Layer Perceptron (MLP), Radial Basis Function (RBF), Self-Organizing Maps (SOM), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) etc. In our approach, we have used RNNs.

In RNN, the information cycles through a loop. When it makes a decision, it takes into consideration the current input and also what it has learned from the input it received previously. But, RNN has short term memory. The RNN takes two inputs, the present and recent past. Recurrent neural network is type of neural network designed for capturing information from sequences/time series data. The recurrent neural network work on the formula is St = fW(St-1, Xt)St is the new state, St-1 is the previous state while is xt the current input and f is recursive RNN. This is shown in figure 5.

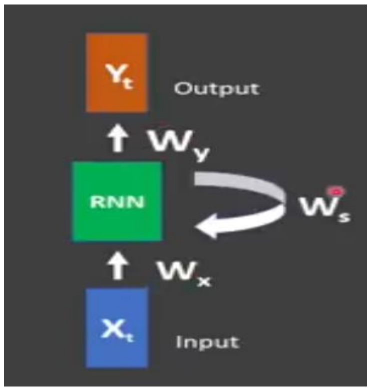

Figure 5: Working of RNN

The recursive function is tanh function the formula is S = tanh (WS St-1 + WxXt) in this St-1 is the previous state multiply by weight (WS) and xt the current input multiply by the Wxweight. So, the new state we get is Yt = WYSt. This is shown in figure 6.

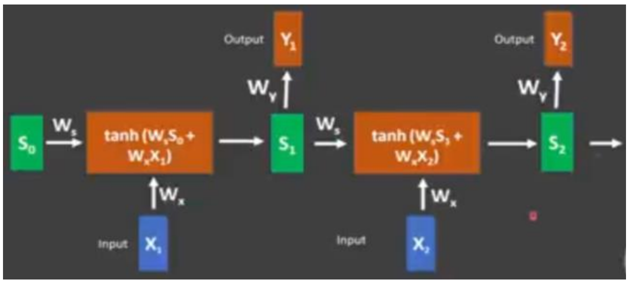

Figure 6: Using tanh activation function in RNN

Figure 7 shows how single layer RNN works. The S0 is the previous state with weight WS and X1 is the current input and new state is calculated by recursive formula tanh (WS St-1 + WxXt) and new state is obtaining St.

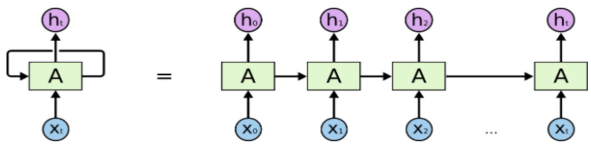

Figure 7: An unrolled RNN

In order to overcome the dependency of short term memory, we have used long short term memory (LSTM). This preserves the information of the previous sequence. Thus, LSTM is special kind of RNN (recurrent neural network) it has long term dependency. To overcome the problem of RNN. It remembers information for long period of time. Due to this LSTM contain its information in a memory of a computer because the LSTM can read, write and delete information from memory. In LSTM memory is seen as a gated sum. Where gate has facility to take decision that which information has to store or delete based on the importance of information. This importance analyses through weight, which are also learned by the algorithm.

LSTM have three gate input, forget and output gate. Input gate determine that coming input is new or not, forget gate determine that information is to be deleted or not and output gate let it impact the output at current time step. Recurrent neural network has the form of a repeating modules of neural network work in chain. In standard RNN has single tanh layer. This is shown in figure 8.

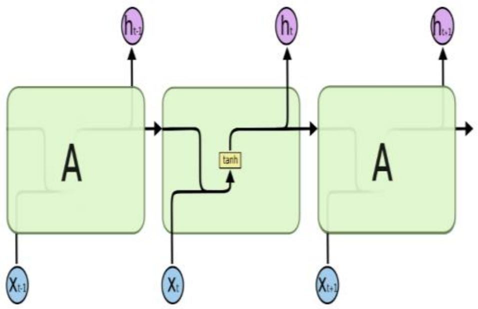

Figure 8: Repeating module of RNN contains single layer

Like RNN LSTM also has chain like structure, but in LSTM repeating module has a different structure. It has four neural network layers, interacting in a very special way. Figure 9 shows this mechanism.

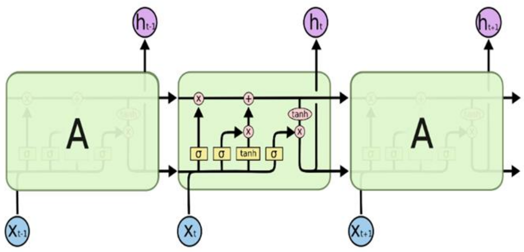

Figure 9: repeating module in an LSTM contains four interacting layer.

In our system, we have performed word and character level embedding and word level embedding in part of speech tagging. word-level embeddings are used to capture syntactic and semantic information, of a sentence and character-level embeddings capture morphological and shape information of a sentence. Figure 10, shows the system architecture of our approach.

Our input sentence be x1,x2………………. xn where xi ∈ v, our vocab(v). Also, let T be our tag set, and yi the tag of word xi. Denote our prediction of the tag of word xi by y^I. This is a structure prediction, model, where our output is a sequence y^1,…,…y^M, where y^I ∈T.For the prediction, pass an LSTM over the sentence. Denote the hidden state at timestep i as hi. Also, assign each tag a unique index. Then our prediction rule for y^i is shown in equation 1.

$$\hat{y}_i = \text{argmax}_j (\log \text{Softmax}(Ah_i + b))_j$$

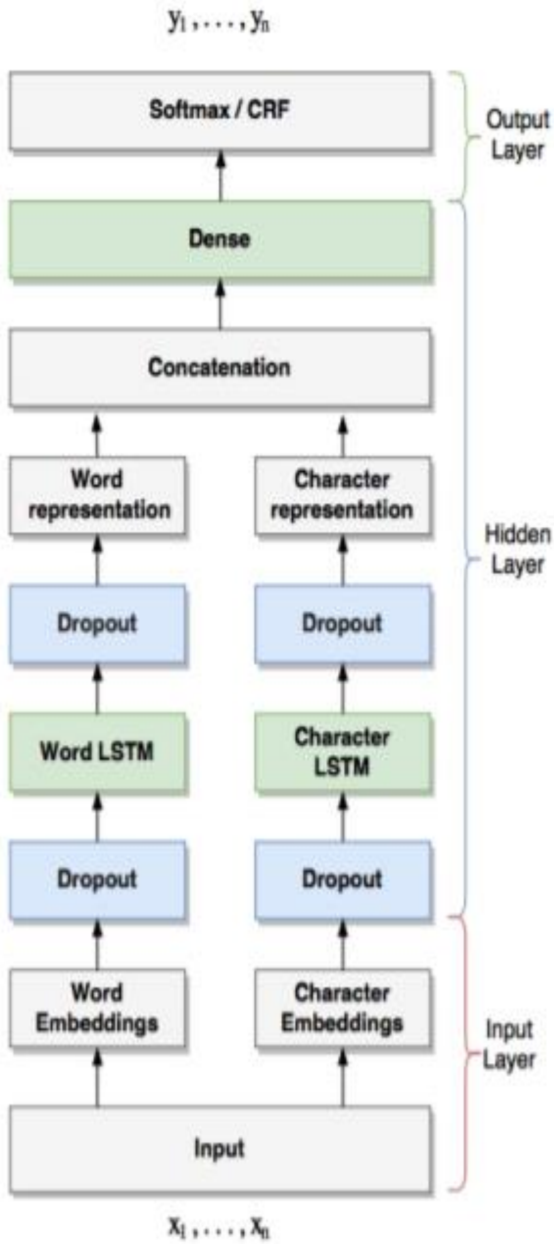

Figure 10: System Architecture

According to neural network layer input layer has word and character level embedding is done. In the hidden layer two LSTMs map both words and character representations to hidden sequences and in the output layer a SoftMax computes the probability distribution over all tag label. So Initially mapping each word in the input to dimensional vectors for sequences of characters and sequences of words. In the hidden layer, the output from the character and word embeddings is used as the input to two LSTM layers to obtain fixed-dimensional representations for characters and words. At the output layer, a SoftMax or a CRF is applied over the hidden representation of the twoLSTMs to obtain the probability distribution over alllabels.

Word embedding:Word embeddings are in fact a class of techniques where individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to one vector and the vector values are learned in a way that resembles a neural network

Character embedding: capture morphological and shape information of a sentence.

IV. EVALUATION

We evaluated our system by generating the output for a given input sentence and then giving to human annotator for manual checking. For this, the accuracy was calculated as follows:

$$Accuracy = \frac{Correct}{Total\ Tags}$$

Where correct means the correct tags identified by the machine for a sentence and total tags means the total tags that were generated by the machine.

This experiment was performed over hind dataset containing 50000 hind tagged sentences. We have trained our tagger on 50000 manually POS tagged sentences. In all these 500 sentences had 6000 words. Out these 6000 words, correctly tag 5823 words. Thus, the accuracy is 97.05%.

V. CONCLUSION

Part of speech tagging is the initial step of natural language processing. In the project we have worked on hind language. As tagging sentence in hind is a challenging task because hind is morphologically riche language. In this no capitalization of starting of the sentence and one word can have multiple meaning. So, to tag this type of sentence is the challenging task. In this project we use 50000 tag sentences and get the accuracy of 97.05%.

REFERENCES

1. Joshi, N., Darbari, H., & Mathur, I. (2013). HMM based POS tagger for Hindi. In Proceeding of 2013 International Conference on Artificial Intelligence, Soft Computing (AISC-2013).
2. Modi, D., & Nain, N. (2016). Part-of-Speech Tagging of Hindi Corpus Using Rule-Based Method. In Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing (pp. 241-247). Springer, New Delhi.
3. Ekbal, A., Haque, R., & 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, 67-78.
4. Dandapat, S., Sarkar, S., & Basu, A. (2004, December). A Hybrid Model for Part-of-Speech Tagging and its Application to Bengali. In International conference on computational intelligence (pp. 169-172).
5. Singh, S., Gupta, K., Shrivastava, M., & Bhattacharyya, P. (2006, July). Morphological richness offsets resource demand-experiences in constructing a pos tagger for hindi. In Proceedings of the COLING/ACL on Main conference poster sessions (pp. 779-786). Association for Computational Linguistics.
6. Dandapat, S., & Sarkar, S. (2006). Part of speech tagging for bengali with hidden markov model. proceeding of the NLPAI Machine Learning Competition.
7. Ekbal, A., Haque, R., & 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).
8. Singh, J., Joshi, N., & Mathur, I. (2013). Part of speech tagging of Marathi text using trigram method. arXiv preprint arXiv:1307.4299.
9. Patil, H. B., Patil, A. S., & Pawar, B. V. (2014). Part-of-Speech Tagger for Marathi Language using Limited Training Corpora. International Journal of Computer Applications, 975, 8887.
10. Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025.
11. Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.
12. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K. and Klingner, J., 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144
13. Sennrich, R., Haddow, B. and Birch, A., 2015. Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909.
14. Luong, M.T., Sutskever, I., Le, Q.V., Vinyals, O. and Zaremba, W., 2014. Addressing the rare word problem in neural machine translation. arXiv preprint arXiv:1410.8206.
15. Tu, Z., Lu, Z., Liu, Y., Liu, X. and Li, H., 2016. Modeling coverage for neural machine translation. arXiv preprint arXiv:1601.04811.
16. Sennrich, R., Haddow, B. and Birch, A., 2015. Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709.
17. Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Barone, A.V.M., Mokry, J. and Nădejde, M., 2017. Nematus: a toolkit for neural machine translation. arXiv preprint arXiv:1703.04357.
18. Joshi, N., Mathur, I. and Mathur, S., 2010. Frequency based predictive input system for Hindi. In Proceedings of the International Conference and Workshop on Emerging Trends in Technology pp. 690-693. ACM.
19. Joshi, N., Mathur, I. and Mathur, S., 2011. Translation memory for Indian languages: an aid for human translators. In Proceedings of the International Conference & Workshop on Emerging Trends in Technology pp. 711-714. ACM.
20. Joshi, N., Darbari, H. and Mathur, I., 2012. Human and automatic evaluation of english to hindi machine translation systems. In Advances in Computer Science, Engineering & Applications pp. 423-432. Springer, Berlin, Heidelberg.
21. Gupta, V., Joshi, N. and Mathur, I., 2013. Rule based stemmer in Urdu. In 2013 4th International conference on computer and communication technology (ICCCT) pp. 129-132. IEEE.
22. Gupta, V., Joshi, N. and Mathur, I., 2013. Subjective and objective evaluation of English to Urdu Machine translation. In 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI) pp. 1520-1525. IEEE.
23. Singh, J., Joshi, N. and Mathur, I., 2013. Development of Marathi part of speech tagger using statistical approach. In 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI) pp. 1554-1559. IEEE.
24. Bhalla, D., Joshi, N. and Mathur, I., 2013. Improving the quality of MT output using novel name entity translation scheme. In 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI) pp. 1548-1553. IEEE.
25. Joshi, N., Mathur, I., Darbar, H., Kumar, A. and Jain, P., 2014. Evaluation of some English-Hindi MT systems. In 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI) pp. 1751-1758. IEEE.
26. Gupta, V., Joshi, N. and Mathur, I., 2014. Evaluation of English-to-Urdu Machine Translation. In Intelligent Computing, Networking, and Informatics pp. 351-358. Springer, New Delhi.
27. Singh, J., Joshi, N. and Mathur, I., 2014. Marathi Parts-of-Speech Tagger Using Supervised Learning. In Intelligent Computing, Networking, and Informatics pp. 251-257. Springer, New Delhi.
28. Joshi, N., Mathur, I., Darbari, H. and Kumar, A., 2015. Incorporating Machine Learning Techniques in MT Evaluation. In Advances in Intelligent Informatics pp. 205-214. Springer, Cham.
29. Tyagi, S., Chopra, D., Mathur, I. and Joshi, N., 2015. Comparison of classifier-based approach with baseline approach for English-Hindi text simplification. In International Conference on Computing, Communication & Automation pp. 290-293. IEEE.
30. Tyagi, S., Chopra, D., Mathur, I. and Joshi, N., 2015. Classifier based text simplification for improved machine translation. In 2015 International Conference on Advances in Computer Engineering and Applications pp. 46-50. IEEE.
31. Yogi, K.K., Joshi, N. and Jha, C.K., 2015. Quality Estimation of MT-Engine Output Using Language Models for Post-editing and Their Comparative Study. In Information Systems Design and Intelligent Applications (pp. 507-514). Springer, New Delhi.

Retrieval Number: A1531109119/2019©BEIESP
DOI: 10.35940/ijeat.A1531.109119
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

32. Gupta, V., Joshi, N. and Mathur, I., 2015. Design & development of rule based inflectional and derivational Urdu stemmer 'Usal'. In 2015 International conference on futuristic trends on computational analysis and knowledge management (ABLAZE) pp. 7-12. IEEE.
33. Asopa, S., Asopa, P., Mathur, I. and Joshi, N., 2016. Rule based chunker for Hindi. In 2016 2nd international conference on contemporary computing and informatics (IC3I) pp. 442-445. IEEE.
34. Gupta, V., Joshi, N. and Mathur, I., 2016. Design and development of a rule-based Urdu lemmatizer. In Proceedings of International Conference on ICT for Sustainable Development pp. 161-169. Springer, Singapore.
35. Kumar, A., Mathur, I., Darbari, H., Purohit, G.N. and Joshi, N., 2016. Implications of Supervised Learning on Word Sense Disambiguation for Hindi. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies. p 54. ACM.
36. Joshi, N., Mathur, I., Darbari, H. and Kumar, A., 2016. Quality Estimation of English-Hindi Machine Translation Systems. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies. p 53. ACM.
37. Chopra, D., Joshi, N. and Mathur, I., 2016. Named Entity Recognition in Hindi Using Conditional Random Fields. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies p. 106. ACM.
38. Chopra, D., Joshi, N. and Mathur, I., 2016. Named Entity Recognition in Hindi Using Hidden Markov Model. In 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT) pp. 581-586. IEEE.
39. Gupta, V., Joshi, N. and Mathur, I., 2016. POS tagger for Urdu using Stochastic approaches. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies p. 56. ACM.
40. Mathur, I., Joshi, N., Darbari, H. and Kumar, A., 2016. Automatic Evaluation of Ontology Matchers. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies. p 56. ACM.
41. Chopra, D., Joshi, N. and Mathur, I., 2016. Improving Quality of Machine Translation Using Text Rewriting. In 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT) pp. 22-27. IEEE.
42. Joshi, M.L., Mittal, N. and Joshi, N., 2017. An Insight into Role of Wordnet and Language Network for effective IR from Hindi Text Documents. In FIRE (Working Notes) pp. 158-163.
43. Singh, S.P., Kumar, A., Darbari, H., Singh, L., Joshi, N., Gupta, P. and Singh, S., 2017. Intelligent System for Automatic Transfer Grammar Creation Using Parallel Corpus. In International Conference on Information and Communication Technology for Intelligent Systems pp. 519-528. Springer, Cham.
44. Singh, S.P., Kumar, A., Darbari, H., Rastogi, A., Jain, S. and Joshi, N., 2017. Building Machine Learning System with Deep Neural Network for Text Processing. In International Conference on Information and Communication Technology for Intelligent Systems pp. 497-504. Springer, Cham.
45. Singh, S.P., Kumar, A., Darbari, H., Kaur, B., Tiwari, K. and Joshi, N., 2017. Intelligent Text Mining Model for English Language Using Deep Neural Network. In International Conference on Information and Communication Technology for Intelligent Systems pp. 473-486. Springer, Cham.
46. Singh, S.P., Kumar, A., Darbari, H., Tailor, N., Rathi, S. and Joshi, N., 2017. Intelligent English to Hindi Language Model Using Translation Memory. In International Conference on Information and Communication Technology for Intelligent Systems pp. 487-496. Springer, Cham.
47. Gupta, C., Jain, A. and Joshi, N., 2018. Fuzzy Logic in Natural Language Processing–A Closer View. Procedia computer science, 132, pp.1375-1384.
48. Gupta, I. and Joshi, N., 2017. Tweet normalization: A knowledge based approach. In 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS) pp. 157-162. IEEE.
49. Gupta, V., Joshi, N. and Mathur, I., 2017. Approach for multiword expression recognition & annotation in urdu corpora. In 2017 Fourth International Conference on Image Information Processing (ICIIP) pp. 1-6. IEEE.
50. Nathani, B., Joshi, N. and Purohit, G.N., 2018. A Rule Based Light Weight Inflectional Stemmer for Sindhi Devanagari Using Affix Stripping Approach. In 2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE) pp. 1-4. IEEE.
51. Asopa, S., Asopa, P., Mathur, I. and Joshi, N., 2019. A Shallow Parsing Model for Hindi Using Conditional Random Field. In International Conference on Innovative Computing and Communications pp. 295-302. Springer, Singapore.
52. Gupta, V., Joshi, N. and Mathur, I., 2019. Advanced Machine Learning Techniques in Natural Language Processing for Indian Languages. In Smart Techniques for a Smarter Planet pp. 117-144. Springer, Cham.
53. Gupta, C., Jain, A. and Joshi, N., 2019. A Novel Approach to feature hierarchy in Aspect Based Sentiment Analysis using OWA operator. In Proceedings of 2nd International Conference on Communication, Computing and Networking pp. 661-667. Springer, Singapore.
54. Sharma P., Joshi N., 2019. Design and development of a knowledge-based approach for word sense disambiguation by using wordnet for Hindi. International Journal of Innovative Technology and Exploring Engineering, Vol 8, pp 73-78.
55. Katyayan, P. and Joshi, N., 2019. Sarcasm Detection Approaches for English Language. In Smart Techniques for a Smarter Planet pp. 167-183. Springer, Cham.
56. Gupta I., Joshi N., 2019. Enhanced Twitter Sentiment Analysis Using Hybrid Approach and by Accounting Local Contextual Semantic. Journal of Intelligent Systems. Walter de Gruyter.
57. Shree, V., Mathur, I., Yadav, G., Joshi, N., 2019. Digital humanities: Can machine translation replace human translation. International Journal of Recent Technology and Engineering. Vol 7, pp 128-134.
58. Ahmed, S.A., Joshi, N., Mathur, I., Katyayan, P., 2019. Impact of related languages as pivot language on machine translation. International Journal of Recent Technology and Engineering. Vol 7, pp 1539-1546.
59. Ahmed, S.A., Joshi, N., Mathur, I., Katyayan, P., 2019. Implications of english as a pivot language in arabic-hindi machine translation. International Journal of Engineering and Advanced Technology. Vol 8, pp 271-277.
60. Seal, S., Joshi, N., 2019. Design of an inflectional rule-based assamese stemmer. International Journal of Innovative Technology and Exploring Engineering. Vol 8, pp 1651-1655.
61. Singh, A., Joshi, N., 2019. Part of speech tagging of Hindi using Markov model. International Journal of Innovative Technology and Exploring Engineering. Vol 8, pp 1723-1726.
62. Pandey, S., Mathur, I. and Joshi, N., 2019. Information Retrieval Ranking Using Machine Learning Techniques. In 2019 Amity International Conference on Artificial Intelligence (AICAI) pp. 86-92. IEEE.
63. Singh, V., Joshi, N., 2019. A rule based approach for anaphora resolution. International Journal of Innovative Technology and Exploring Engineering. Vol 8, pp 2652-2657.
64. Sinha, S., Saxena, K., Joshi, N., 2019. Sentiment analysis of facebook posts using hybrid method. International Journal of Recent Technology and Engineering. Vol 8, pp 2421-2428.
65. Kumar, S. and Joshi, N., 2016. Rule power factor: a new interest measure in associative classification. Procedia Computer Science, 93, pp.12-18.
66. Sharma, O., Kumar, S. and Joshi, N., 2019. SRPF Interest Measure Based Classification to Extract Important Patterns. In Proceedings of 2nd International Conference on Communication, Computing and Networking pp. 523-530. Springer, Singapore.
67. Goyal, H., Sharma, C. and Joshi, N., 2017, August. Estimation of Monthly Rainfall using Machine Learning Approaches. In 2017 International Conference on Innovations in Control, Communication and Information Systems (ICICCI) pp. 1-6. IEEE.
68. Goyal, H., Joshi, N. and Sharma, C., 2019. Feature Extraction in Geospatio-temporal Satellite Data for Vegetation Monitoring. In Emerging Trends in Expert Applications and Security pp. 177-187. Springer, Singapore.
69. Màrquez, Lluís, and Horacio Rodríguez. "Part-of-speech tagging using decision trees." European Conference on Machine Learning. Springer, Berlin, Heidelberg, 1998.
70. 25. Zhang, Yuan, et al. "Learning Tag Dependencies for Sequence Tagging." IJCAI. 2018. Vaswani, Ashish, et al. "Supertagging with lstms." Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016.
71. 26. Tan, Tien-Ping, et al. "Evaluating LSTM Networks, HMM and WFST in Malay Part-of-Speech Tagging." Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 9.2-9 (2017): 79-83

72. 27. Perez-Ortiz, Juan Antonio, and Mikel L. Forcada. "Part-of-speech tagging with recurrent neural networks." Universitatd'Alacant, Spain (2001).
73. 28. Huang, Zhiheng, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging." arXiv preprint arXiv:1508.01991 (2015).
74. 29. Zheng, Xiaoqing, Hanyang Chen, and Tianyu Xu. "Deep learning for Chinese word segmentation and POS tagging." Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013.

Retrieval Number: A1531109119/2019©BEIESP
DOI: 10.35940/ijeat.A1531.109119
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication