Part of Speech Tagging for English Language using Bidirectional Long Short Term Memory Recurrent Neural Network

Ahmer Sabah
Student Id: 1094243
Lakehead University

Alizar Marchawala
Student Id: 1103258
Lakehead University

Harishma Tharakkal Haridas
Student Id: 1114334
Lakehead University

Vinit Krishnankutty
Student Id: 1096016
Lakehead University

***Abstract*—Parts-of-speech (POS) Tagging is a method of labelling a term in a corpus, depending on its meaning and description, into a corresponding part of a speech tag. This function is not simple, as a single word may include another component of expression depending on the context in which the word is used. POS Tagging is really an important step for interpreting the context of every sentence or for extracting relationships and constructing an information graph. Bidirectional Long Short-Term Memory (BLSTM) is very efficient in marking sequential data. In this paper, we proposed BLSTM for the POS tagging task. We use a Recurrent Neural Network (RNN) with BLSTM to study the proper choice of words based on a sentential sense of a word. When tested on a test set, 92.0 tagging accuracy and 0.03 loss is achieved. This method can also achieve a decent score relative to the standard POS tagger, without using any additional features.**

***Index Terms*—POS Tagging, BLSTM, RNN, CNN, HMM, LSTM, CRF**

I. Introduction

POS Tagging is the process of identifying and marking a word. This marking is done as part of identifying a word meaning with respect to part of speech. This process purely depends on the definition and the context of the word in the sentence. POS is usually defined as the primary step to analyse any data syntactically. Different kinds of parts of speech in the English Language are noun, pronoun, adjective, determiner, verb, adverb, preposition, conjunction and interjection. Consider the sentence "The Blue Can". The different parts of speech for this sentence are determiner, adjective, and noun. A system is to be developed that could tag any sequence of words. There are different kinds of POS tagging techniques such as lexical based methods, rule-based methods, probabilistic methods, and deep learning methods. Lexical based methods will assign the most frequently occurring POS tag for a phrase in the training corpus. Rule-based methods select tags based on rules. If we are generating a rule which says, words ending with "ing" must be assigned to a verb. Along with Lexical Dependent methods, these strategies may be used to allow POS tagging of terms that are not in the training corpus but are in the test results. The probabilistic approach assigns the POS tags based on the occurring probability of a specific tag sequence. Conditional Random Fields (CRFs) and Hidden Markov Models (HMMs) are different probabilistic methods to assign a POS Tag. Deep Learning Methods will use RNN for POS tagging.

Long short-term memory (LSTM) is a RNN model used in deep learning. LSTM is able to learn long-term dependencies and work remarkably well on a wide variety of problems that are now widely used. It can remember information for long periods of time. A RNN is a class of artificial neural networks where node-to-node connections form a direct graph along a sequence of time. This makes temporary complex behaviour. RNNs derived from feedforward neural networks may use their internal state to process input sequences of variable lengths. BLSTM is used as an algorithm for the retrieval of sequential data. This supervised learning approach trains a specific RNN utilizing a mixture of nonlinear processing elements and linear feedback loops to store long-range information to use a very long-range symmetric sequence background. BLSTM is a category of RNN that can integrate long-term fore-and-aft input contextual information. It is a strong model for sequential labelling activities which has been proved. Every word is converted into its dense embedding representation and then fed into two LSTM networks such as left-to-right LSTM network and right-to-left LSTM network. The outputs of the two hidden states of the LSTM are concatenated further, and the analysis is carried out. A model learns from the prefix and suffix meanings of the target term when applying this approach to the English language, and performs the final classification based on the jointly learned representation of this meaning.

Word vectors are basically a much more effective way of representing words. Word vectors take up far less space than one-hot encoded vectors and carry semantic knowledge about the word as well. These are the main features of word vectors. Google's Word2Vec is among the first implementations of word vectors. There are basically two word2vec versions that are skip-gram and CBOW. Word2vec offers direct access to vector representations of words that can assist to obtain decent performance over a number of tasks that machines are traditionally not good at. Skip-gram tries to forecast a term to the immediate neighbors. It looks like a word ahead,

a word behind, or two left or some other combination. We apply the word that we want to model as our input X and the corresponding words as the output Y target.

II. Literature survey

Recognition of emotions in the text is performed using LSTM [2] with the help of vector of words which is semantic and emotion oriented. Word2vec model is considered, from which the word vector which is semantic of each word from the text inputted to the model is taken out. In order to obtain the emotions associated with each word vector, the lexical word is highlighted to words that indicate the emotions. Then the process of reducing the dimensionality is performed on the word vector which indicates the emotions. This is carried out using the auto encoder by identifying the features which are bottleneck. Bottleneck features are the features which indicate minimum number of neurons that are combined to perform reduction in the number of features. In order to frame features of textual representation in the conclusion stage for the recognition of the emotions, features which are in the bottleneck that are identified by the auto encoder is merged with the word vector features indicating the semantics. Considering the text which is fed into the model for the recognition of the emotion is modelled in terms the context, recognition of emotions is carried out by LSTM. This is carried out based on the entire sequence of feature in the form of text. The various emotions used for the training are anxiety, surprise, boredom, happiness, sad, anger and disgust. The validation technique used for the evaluation of the model is cross validation which is five fold. The research work put forward the chance of improving accuracy based on increasing the size of the dataset used.

The rapid growth of neural networks and deep learning [3] making use of multiple hidden layers, made the usage of Convolutional Neural Network (CNN) and LSTM in broader aspects. A research work is carried out based on using CNN and LSTM in the combined manner to perform the classification of text in an effective manner. This performs categorization of text based on both object and the subject. The categorization of text implemented using LSTM and CNN make use of four layers. The layers are layer for accepting the input, convolution layer, LSTM layer and various variants of that layer and the classifier layer which is carried out using probability distribution known as softmax layer. The data fed into model is based on both object and subject of the words. The data is pre-processed and the vector of words are created. Convolution layer is defined as an activation function layer which has the nonlinear properties. A filter also known as the kernel is created which helps in the feature extraction by sliding the filter through all parts of the input data. LSTM is referred to as a variety or special instance of RNN. The LSTM keeps cell state, which can perform the addition and deletion of data to the different cells. The three types of gates that LSTM make use are namely input, forget and the final gate which is output gate. Hence CNN performs the extraction of features are high level. The output created is confined to 0 or 1 by the final activation layer which is softmax. This research suggests the idea of more accuracy in terms combination of CNN without the activation function and the LSTM.

The internet is flooding with huge amount of information. Hence the need of an apt tool for harnessing the online data is highly essential. A technique based on LSTM RNN [4] make use of the machine learning ability to extract the data. This technique provides the easy way of getting the results of search based on the context the user wants. The technique introduced helps to perform the mechanism of crawling with minimum latency high efficiency search based on the context of words in a sentence. The proposed system has an architecture referred to as a stack MEAN, where M stands for MongoDB, E is the abbreviation for Express.js, A stands for AngularJS, and N indicates NodeJS.

RNN is referred to as the most efficient architecture in Deep Learning for Natural Language Processing [4]. This is mentioned considering the ability of RNN to make use of the previous steps based on time. The research indicates the comparative study on three different types of RNN namely Vanilla, LSTM and Gated. The dataset used in the research is the Google news. The result of the research indicates that Gated RNN results with the best accuracy on datasets for sentiment analysis. In order to learn the complete emotions from the POS words [5], two models are proposed. Both the models make use of LSTM, with the first model considers the vectors from POS tags and chain them with the hidden layers of the network. This generates the mechanism for attention to create representations for the text with respect to each emotion. The second model is combining attention mechanism with LSTM for the representation of words in terms of their contexts and combine them as the representation of text for every emotion.

III. Proposed Method

Here, we propose using RNN based BLSTM to predict tags for each word. LSTM receives input and gives out corresponding output along with updating its internal hidden state simultaneously. This mechanism is similar to HMMs except that here we use neural networks.

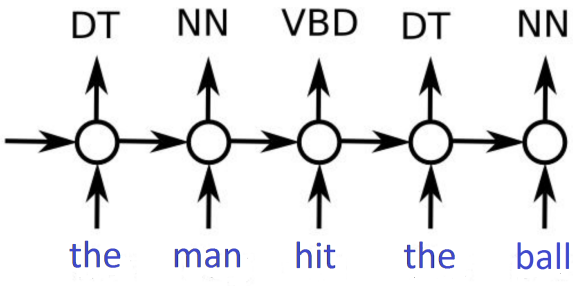

Fig. 1. POS Tagging [7]

Using BLSTM gives us the flexibility to feed the network with more amount of input information. LSTM stores only

that information that has passed through it using hidden state. Using BLSTM will allow our input to run in two directions, one from previous step or past to next step or future and vice versa. This way we can store information from both past and future using two hidden states. This way we can develop better understanding of the context. We then perform vector representation for each word in the corpus as this gives more semantic and syntactic information about any word. The data is further processed before being fed to the neural network. Data processing has two steps.

1) Padding: We pad all the words with zeroes in the beginning to make all the sentences have same dimension. This reduces the training time of the model.
2) Converting tags to categorical form: The trained network gives us the predictions of probabilities of possible tags for each word. These tags should be represented as N dimensional vector , where N is total number of tags. Here, we use one-hot encoding to represent categorical variables as binary vectors.

Once the data is ready, we train our RNN to perform POS tagging.

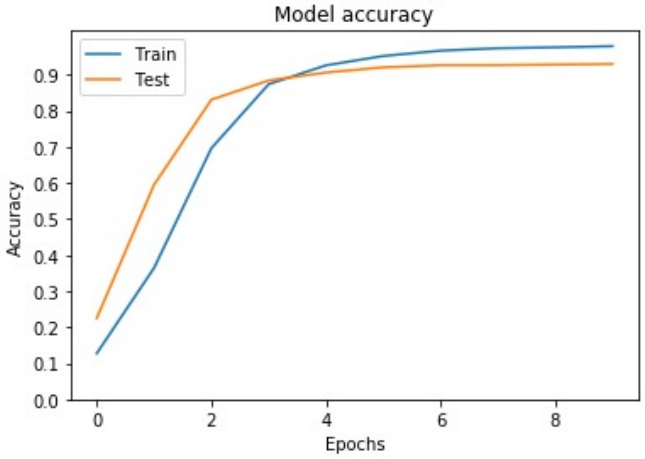

Fig. 2. Model Accuracy

IV. Experimental Analysis

For the experiment, we used dataset from Penn Tree Bank. We have implemented the task on Kaggle platform using Python on Keras which is a wrapper for the TensorFlow library by Google. Training datasets and testing datasets are split in the ratio of 80:20. The data is prepared to be trained. The words are converted to lower case and are padded with zeroes. This prevents creation of staggered matrix/tensor and reduces training time.Then the words are converted to their respective vector representation using word2vec model. Gensim library will enable us to develop word embeddings by training our own word2vec models on any corpus either with CBOW of skip-grams algorithms.

Gensim word2vec requires 'list of lists' format for training where every document is stored in the form of a list and every list contains lists of tokens of that document. So, we create lists of words and tags.We then define our word2vec model with embedding dimensions size as 300. We used skip-gram model as a training algorithm. The maximum distance between a target word and words around the target word is known as window. We set our window size as 7. The model also considers minimum count of words while training the model; words with occurrence less than this count will be ignored. We set this as 1. We set cpu_count() as our number of partitions in order to efficiently utilize our system resources. Hierarchial softmax is used for model training. We can set two learning rates, one would be the initial value and the other would the value until which the learning rate can linearly drop. We have trained the model on fixed learning rate at 0.01. We create embedding matrix to form dense representations of words and their meanings. We create a sequential BLSTM network to train our model. Our network has one embedding layer to encode the statements with indices with dimension of 300. Since, we have padded our data we set the mask_zero value to True. This is useful while using recurrent layers with variable length input. The subsequent layers support masking of data as index 0 cannot be used in the vocabulary built. We then add BLSTM layers to our network. These layers analyzes the pattern in the given data sequences and outputs feature patterns making it easy to understand for our subsequent layers to perform prediction on the data.

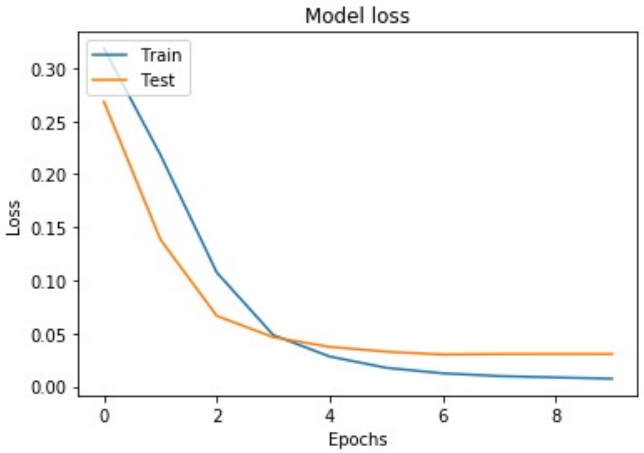

Fig. 3. Model Loss

In order to train our model to handle any mathematical function, we introduce some non-linearity property to it by adding dense layers. We added TimeDistributed wrapper to our dense layer to maintain one-to-one relationship between the input and output. To prevent our model from over-fitting we added dropout layer. This acts as a regularization method where the outputs of layers under dropout are sampled randomly thereby reducing overfitting and improving performance of the model. The output is then passed through a non-linear function i.e., softmax activation function which converts the vectors to

probability values.

$$\sigma(X_j) = \frac{e^{x_j}}{\sum_i e^{x_i}} \quad (1)$$

The tag with highest probability is considered as the output tag. We reduce the classification error of the network using Categorical Cross Entropy as loss function and ADAM optimizer to reduce the defined loss.

V. Comparison Results

We tried the experiment with different number of epochs and optimizing algorithms ADAM and NADAM. However, both the algorithms almost gave similar performance whereas SGD optimizer could not give significant results for the same. It was also noted that reducing the batch size from 256 to 128 to 64 drastically impacted the accuracy the model. Our model gave 92.6% accuracy with learning rate at 0.01.The accuracy and loss graphs plotted during the experiment can be seen in the figures 2 and 3 respectively.

TABLE I
Comparison result

Optimizer	Learning Rate	Accuracy	Loss
ADAM	0.01	92.96	0.041
NADAM	0.01	92.6	0.03
ADAM	0.001	87	0.11
NADAM	0.001	88	0.047

VI. conclusion

We proposed BLSTM RNN for POS tagging. Combined with word embedding trained on unlabeled data, this approach gets good accuracy on our test set.Use of word embeddings with BLSTM RNN has proved to be reliable and efficient method to perform POS tagging and also gives scope for further exploration.

This experiment was re-conducted as a group effort. Each member have equally contributed towards it from its inception till the completion.Each member worked towards gathering source information, literature work ,coding and execution and report preparation.For the project's 100% completion, each member has participated fair and equally at 25%.

References

[1] Peilu Wang, Yao Qian, Frank K Soong, Lei He, and Hai Zhao. 2015. Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. arXiv preprint arXiv:1510.06168.

[2] Ming-Hsiang Su, Chung-Hsien Wu, Kun-Yi Huang, Qian-Bei Hong," LSTM-based Text Emotion Recognition Using Semantic and Emotional Word Vectors", 2018 First Asian Conference on Affective Computing and Intelligent Interaction (ACII Asia).

[3] Yuandong Luan, Shaofu Lin," Research on Text Classification Based on CNN and LSTM", 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA).

[4] Mohammad Arifur Rahman, Fahad Ahmed, Nafis Ali," Contextual Deep Search using Long Short Term Memory Recurrent Neural Network", 2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST).

[5] Kiran Baktha, B K Tripathy," Investigation of Recurrent Neural Networks in the field of Sentiment Analysis", International Conference on Communication and Signal Processing, April 6-8, 2017, India.

[6] Kai Gao, Hua Xu, Chengliang Gao, Hanyong Hao, Junhui Deng, Xiaomin Sun" Attention-Based BiLSTM Network with Lexical Feature for Emotion Classification", 2018 International Joint Conference on Neural Networks (IJCNN).

[7] https://www.stackoverflow.com