Part-of-Speech Tagging classification with Artificial Neural Networks

Adel F. del Valle[1], Guillermo Betancourt[2], Blas Ayala[3], Yeniel Díaz[4]

Advisors: Dr. Fernando Vega Riveros , Dr. Nayda G. Santiago, Hilton Alers Valentín

Research Group of Computational Linguistics
Department of Electrical and Computer Engineering
Department of Hispanic Studies
University of Puerto Rico , Mayagüez Campus
PO Box 9000
Mayagüez, PR 00681-9000

Abstract

Current Part-of-Speech taggers accuracy lies in a asymptote difficult to cross, as they face problems handling ambiguity and information they have not seen. Therefore, the approach suggested in this project considers linguistic principles in the design of the stochastic model. The methods to follow consists of the experimentation with a Deep Feedforward and Bidirectional LSTM networks, comparing how Long-Short Term Memory layers aids the system with a track of time, each word acting as a state. After the training concluded, the network reflected a training accuracy of 99% and a validation accuracy of 97%. The results obtained validates our approach and allows it to be built upon for further experimentation.

I. Introduction

Since the last century, computer scientists and linguists have worked together to develop part-of-speech taggers. Great efforts has been invested with incredible results. Nevertheless, the models tend to an asymptote difficult to push up. Christopher D. Manning talks about how recent Machine Learning tendencies have showed to be efficient, but there is a limitation within the linguistic corpora used as it contains errors that are learned by the model itself [1]. It is an important task to reduce the margin error as possible because a small percentage as 3% holds for thousands of mislabeled words. Consequently, a single mistake affects the interpretation of further analyses. For instance, documents as contracts are constantly being analyzed with algorithms to resolve legal fights where ambiguity is involved. This proper application in courts save time and reduce prone to human errors as it takes a linguist to analyze thousands of words in depth.

In this approach, I propose an Artificial Neural Network model capable of handling ambiguity through Long-Short Term Memory states. The model is designed to receive sentences encoded in sequences through an embedding layer that reduces the dimensionality and masks the padded zeroes. Afterwards, the output of the embedding layer goes into a bidirectional Long-Short Term Memory (LSTM) layer responsible of keeping a sense of time, each word acting as a state. This property allows the network to classify a word in respect to the words that come before and after, applying the principle of syntactic distribution [2]. Finally, the output of the LSTM is applied to a Time Distributed function with a softmax activation function and 12 output neurons. Each neuron represent a respective part-of-speech tag.

The training consisted of 10 epochs and batches of size 512. Each epoch took approximately 10 minutes to train on a computer with NVIDIA GeForce 1060 GTX, 1280 CUDA Cores, Intel Core i7 8th Geneneration and 16GB of RAM.

The computation time in Big-O notation is derived in the following formula. In addition, the model is implemented in Python using TensorFlow, Keras, NumPy and Natural Language Tool Kit (NLTK) frameworks [3]. The respective variables consist of n amount of epochs, t training samples, $\{i, j, k, l\}$ amount of neurons in each layer, and b as the bias [4].

Equation 1. $O(nt(jk+kl)) + O(nt(ij + kl) + b$

This paper consists of three main parts. Section two explains the methodology followed to set up the experiment and the respective variables in a quantitative approach. Section three lays put the results with a respective analysis with relevant results in literature. Section four discusses the conclusions with the respective future work.

II. METHODS

Part-of-Speech (PoS) current taggers are quite efficient. However, accuracies drop markedly when there are differences in topic, epoch, or writing *Table 1. Data sets*style between the training and operational data [1]. For the purposes of this research, the recollected data used is *Brown University Corpus*. The respective texts were preprocessed to build a vocabulary with an integer representation for each token. Alongside, data engineering and featuring was applied, selecting sentences as samples and tags as classes. The sentences were transformed into sequences using the built vocabulary integer representations. In addition, each class was converted to a one-hot encoded vector. The resulting arrays were padded to have sequences of the same length and divided into training, validation and testing data sets as shown below.

Table 1. Data sets

Set	Tokens	Sentences	Shape	Data Type
Training	834, 429	40,000	(40000, 180,)	NumPy Array
Validation	118, 831	10000	(10000, 180,)	NumPy Array
Testing	73, 430	5000	(5000, 180,)	NumPy Array

In the implementation and design of the architecture for the network, a Deep Feedforward model was built with an input shape of (180,), an embedding layer with masking properties and a dense output layer of shape (12, 0). Each output vector represents a tag label. The model was trained with batches of size 512 and 8 epochs; the loss was calculated with Categorical Cross Entropy function and used Mean Squared Error as optimizer. The previously described architecture ended up with a training and validation accuracy of 95% and 94%. Based on relevant literature as Christopher D. Manning’s paper, the results obtained by the Deep Feedforward architecture were not enough to be in a range of the accuracy of a good tagger. Therefore, a Bidirectional LSTM model was proposed to handle ambiguity in sentences and maintain a sense of memory by each token taking a form of state. The respective design consisted of a input layer with 180 units, followed by an embedding layer along with LSTM (128 units) and Time Distributed (12 units) layers. The training parameters are the same as the Deep Feedforward Model. The final architecture is shown below with its respective properties as activation functions and dimensionality.

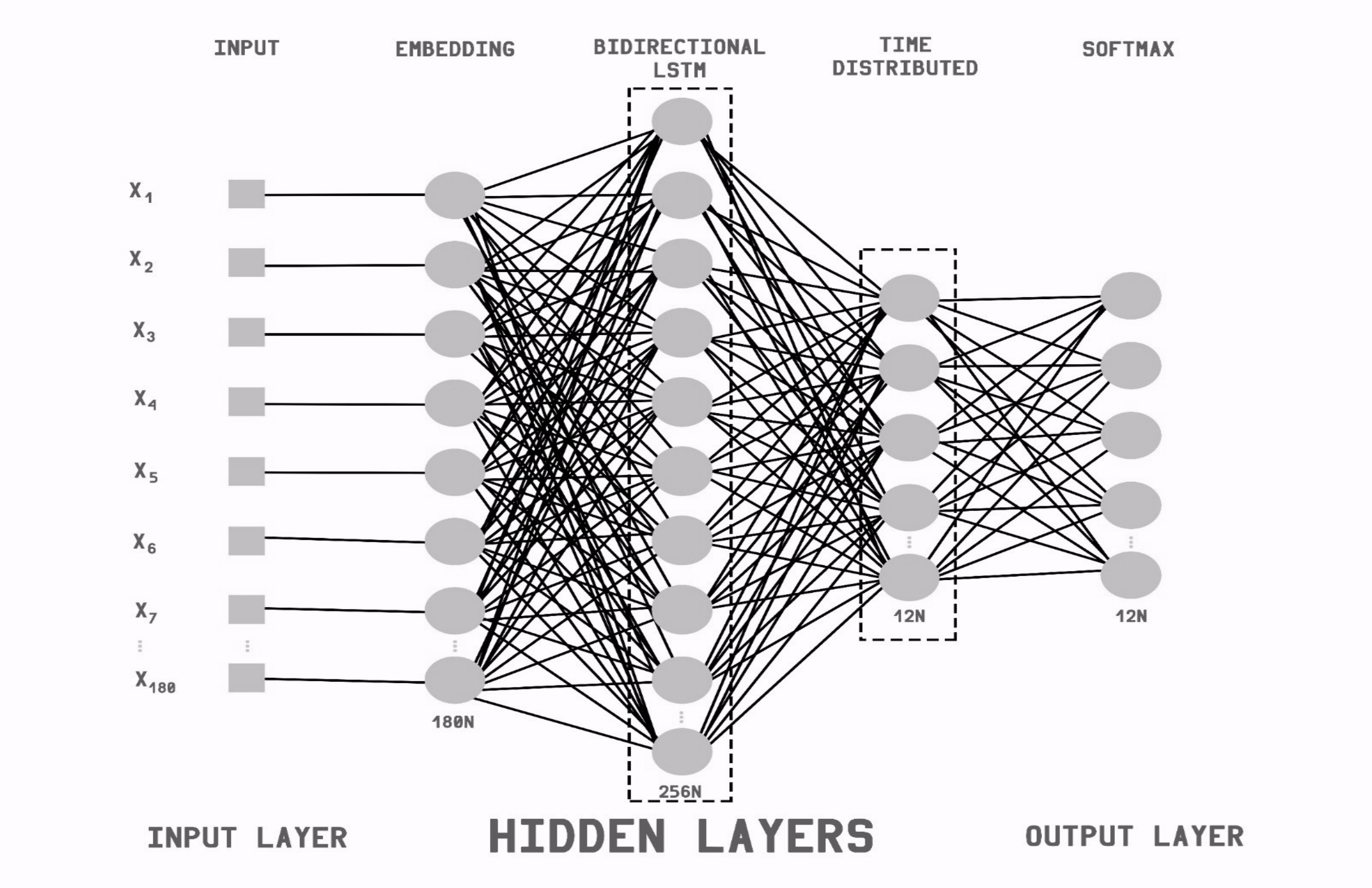

Figure 1. Architecture of Bidirectional LSTM model.

During the experiment, multiple set backs were encountered. For example, working with raw text and the process of transforming it to the way the network receives it was a challenging task. The first problem was a memory allocation issue arisen by the high dimensionality of the returned arrays. At first, the embedding used to represent the vocabulary was Gensim Word2Vec [5] ,which creates a vector for each word based on the cosine similarity with near words. After the pre-processing, the arrays ended up with a shape of (40000, 180, 100). Consequently, any process applied to the arrays consumed a lot of memory inefficiently. It was resolved by representing words with integers and adding an embedding layer to the architecture of the network. Also, the model faced problems during the training because the dimensionality of the layers and the given sets were not matching. Once again, the preprocessing functions were optimized to end up with arrays of the same shape.

[1] https://github.com/NLP-Neural-Network/MLP-Recurrent-NN

III. RESULTS AND ANALYSIS

The bidirectional LSTM model training consisted of 8 epochs and batches of size 512. Once the training was completed, the following results were obtained in terms of training, validation and testing accuracy.

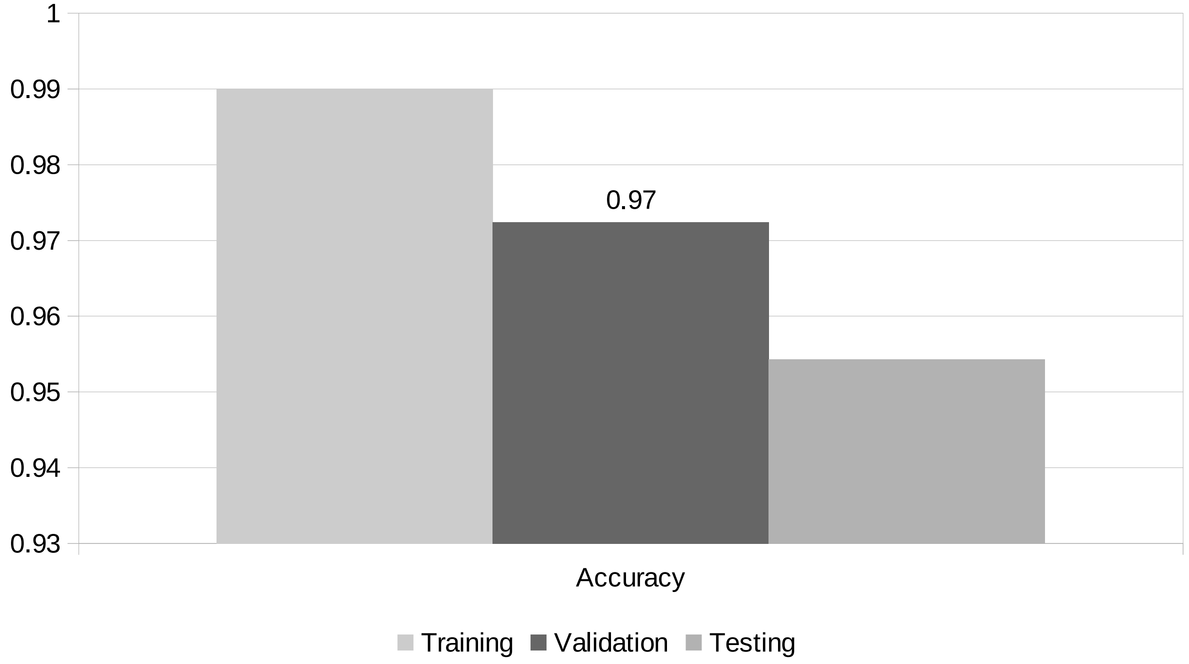

Chart 1. Each data set accuracy percentage

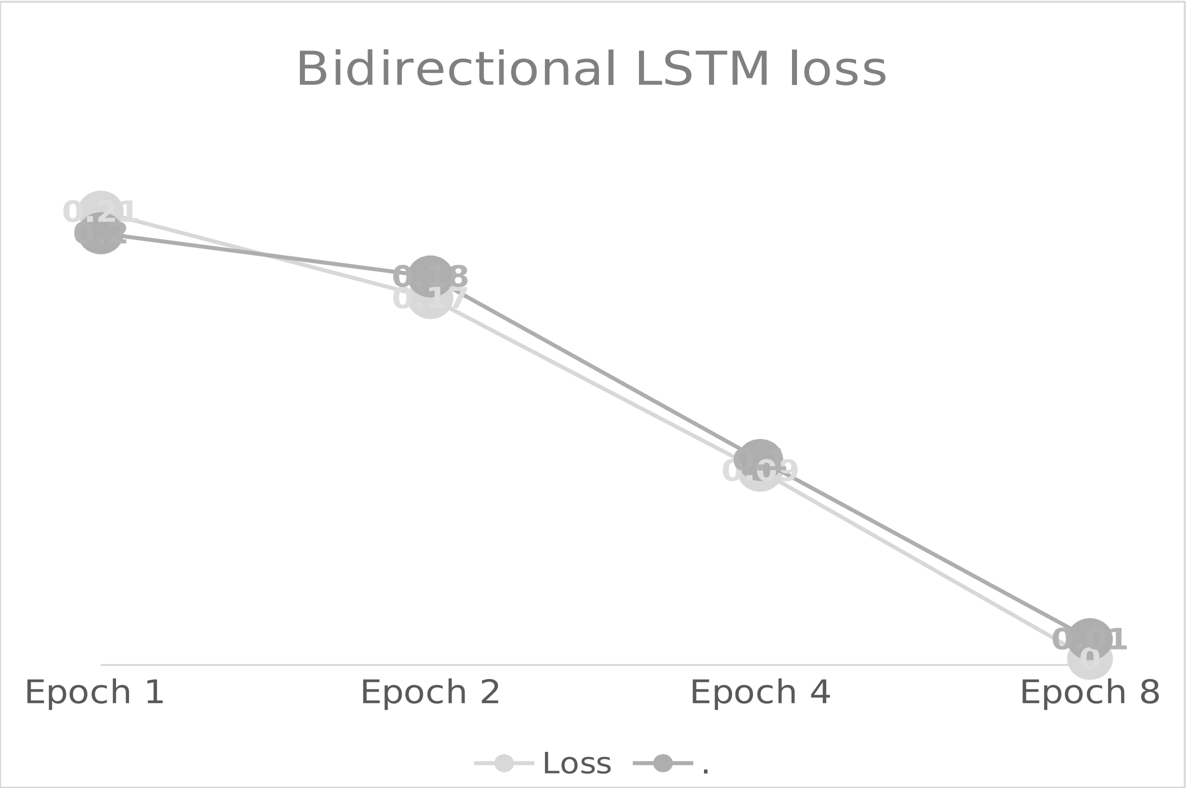

Chart 2. Loss during training per epoch.

The presented percentages validate this approach as it pertains to relevant literature: the accuracy lies within the range to be considered a good PoS tagger. It is no surprise as Long-Short Term Memory have showed to be efficient in similar tools of Natural Language Processing (NLP). It is important to remark that Brown University Corpus was used as data set for the purposes of this project. Language is an infinite and recursive set of elements, accordingly the model respective accuracy is limited to the vocabulary seen during the training.

IV. Conclusion and future work

In comparison with other approaches, the presented model takes in mind basic linguistic principles and does not relies only on the probabilistic aspects. The given results are promising in the development of Part-of-Speech taggers and Natural Language Processing tools, as raw text analyses keep growing exponentially every day. The design of the experiment and the steps taken showed to be fruitful. Alongside, the implementation of the same presented multiple challenges to the team as processing and working with Natural Language on computers is not an easy nor intuitive task. Consequently, the principal lesson learned is that handling data, specially text, it takes a lot of patience and introspection of the methodology one is approaching to manipulate the data. As you go further in the process, in each step you have a problem and it summarizes a constant line of thought of: "How can I solve it?". The future work consists of implementing other corpora, improve the pre-processing functions to allow an analyses of words out of the vocabulary, and better handling of ambiguity in homonyms that can possess different categories along the utterance. The project can be found as an open source project[1] on GitHub.

V. Bibliographic References

[1] CC. Manning, "Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?" CS and Linguistics Dpt., Univ. Dpt., Univ. Stanford, California, 2011.

[2] Y. Abe, "Semantic Categories and Syntactic Distribution", Rikkyo University, Tokyo, Japan, 1983.

[3] F. Chollet, *Deep Learning with Python*. Connecticut, MA: Manning Publications, 2017.

[4] H.Sak, A. Senior, F. Beaufays, "Long Short-Term Memory based recurrent neural network architectures for large vocabulary speech recognition" ICASSP, 2014

[5] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed Representations of Word and Phrases and their Compositionally", Google Inc., California,

VI.

Authors

[1]First Author – *Adel F. del Valle*, Undergraduate Research student, University of Puerto Rico, Mayagüez, adel.delvalle@upr.edu

[2]Second Author – *Guillermo Betancourt*, Undergraduate Research student, University of Puerto Rico, Mayagüez, guillermo.betancourt@upr.edu

[3]Third Author – *Blas Ayala*, Undergraduate Research student, University of Puerto Rico, Mayagüez, blas.ayala@upr.edu

[4]Four Author – *Yeniel Díaz*, Undergraduate Research student, University of Puerto Rico, Mayagüez, yeniel.diaz@upr.edu

[1] https://github.com/NLP-Neural-Network/MLP-Recurrent-NN