POS Tagging Using Transformer Architecture

Vishisht Rao
Computer Science and Engineering
PES University
Bangalore, India
vishisht.rao@gmail.com

Rithvik G
Computer Science and Engineering
PES University
Bangalore, India
rithvik.ganeshhs@gmail.com

V R Badri Prasad
Department of
Computer Science
PES University
Bangalore, India

***Abstract*—Part of Speech(POS) is a contextual categorical tag given to a word wherein words with the same POS tag contain similar grammatical and linguistically structural properties and can often replace one another in a sentence or phrase while maintaining the syntactic integrity of that sentence or phrase. Some common POS tags are noun, adjective, adverb, pronoun, verb, etc. POS Tagging is the process of assigning a POS tag to a word in any given context. Transformer architecture is an augmented form of the encoder decoder model in neural networks which brings about the concept of attention mechanism to give higher importance to certain parts of the input when compared to the other parts. This paper shows how POS tagging can be performed through the use of the transformer architecture and how it measures up to other commonly used text classification models.**

Keywords—BERT, NLTK, Transformer, Encoder, Part Of Speech

I. Introduction

The POS tag given to a word in a particular context gives a lot of grammatical structure to that word. Some of the commonly occurring POS tags are noun, verb, adjective, proposition, etc. Consider the sentence "I am playing *football*." The word "football" in this sentence is a noun, if this word were to be replaced by another noun such as "cricket", the sentence would become "I am playing *cricket*." Both these sentences obey all the syntactic laws of grammar however the two sentences represent very different meanings. Here the words "football" and "cricket" paint a very different picture in one's mind while visualizing the sentence even though only a single word has changed and shows the importance and power a group of words can hold in performing various tasks and analyses.

POS tagging is a very well known task in natural language processing and involves the assignment of a POS tag to the words in a given sentence or phrase. It has a wide variety of applications from simple ones like Named Entity Recognition(NER) and Word Sense Disambiguation to more high level concepts such as sentiment analysis and question answering. To construct such applications it is vital that one has a highly accurate POS tagging model as that can ensure the best possible performance of any application that uses it.

One of the early methods to perform POS tagging in machine learning was by initialising a mapping for each word to its possible POS tags and then further filtering the possible tags down to just one tag through the use of handwritten disambiguation rules. This method is known as rule based POS tagging, it requires immense domain knowledge and it is impossible to determine all possible disambiguation rules. Another method is the stochastic method which assigns a POS tag on a word based on the probability that that word may have that tag and the frequency with which that word has occurred with that tag. Even though this is simple and efficient, it will not yield accurate results for unseen words or rare contextual settings of words. One of the more recent methods to perform POS tagging is through variations of Recurrent Neural Networks such as GRUs or LSTMs. These have shown to yield highly accurate results.

An encoder decoder architecture in neural networks is one which first encodes a sequence of inputs to an intermediary form and then decodes this form to a sequence of outputs. Some applications of encoder decoder models are generating captions for a given image, translation from one language to another, chatbots which are capable of holding conversations, etc.

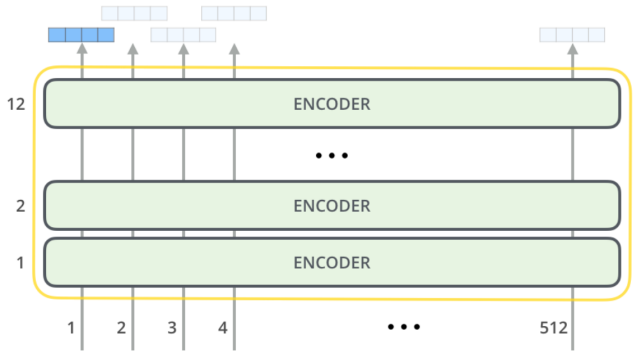
Transformers are a variant of the encoder decoder architecture which gives more attention to certain parts of the input sequence as compared to the other parts. A simple understanding of this idea would be to consider the sentence "The boy is playing with the ball." The meaning of this sentence is stressed in the words "boy" and "ball" along with the word "playing". The words "the", "with" and "is" do not need to be given as much importance or attention.

The first step to any text processing task is to convert the text into a format which can conveniently be input to any model, i.e. a vector also known as a word embedding. Word embeddings were initially computed using simple stochastic methods until 2013 when Word2Vec was released by a team at Google. While Word2Vec provided a steep increase in performance over its predecessors, it was not contextualized. Word2Vec was followed by the release of ELMo which provided contextualized word embeddings for a given sequence of words, however this was not strictly bidirectional.

BERT which stands for Bidirectional Encoder Representations from Transformers is a neural network architecture which consists only of the encoder part in the transformer architecture. Google released the transformer architecture in 2017 and released the BERT model in 2018. The BERT model outputs contextualized word embeddings given an input sequence of words. It has been trained on a massive amount of unlabelled text which includes the entire Wikipedia(over 2,500 million words) and the Book Corpus(over 800 million words). BERT is said to be bidirectional as it processes the input text from left to right as well as right to left through its attention layer, which performs attention in both directions.

The image below shows the diagrammatic representation of the small BERT model introduced by Google which will be used in this paper. This model consists of twelve hidden

layers, 512 neurons per hidden layer and outputs vectors of size 768 for each input in the sequence.



This paper shows how POS tagging can be done through the contextualized word embeddings obtained from the BERT model by fine tuning it and makes comparisons with some standard libraries such as nltk on untagged sentences.

II. Related Works

The authors of [1] use BERT and XLNet, 2 transformer RNNs to obtain contextual word embeddings trained on a subset of Educational Testing Service(ETS) corpus of written English that is non-native. The author then proceeded to detect metaphors present in a given sentence or phrase using these word embeddings and trained it on the VU Amsterdam Metaphor Corpus dataset. In this paper it is shown that contextualized word embeddings perform better than previous word embedding benchmarks for metaphor detection. F1 scores of 0.642 and 0.608 were achieved on 80/20 training-test split of VUA and TOEFL respectively. In order to perform the POS tagging, contextualized word embeddings need to initially be obtained. The intention in this paper is to use transformers, more specifically BERT, for this particular task.

The authors of [2] focus on developing a model for the POS tagging problem based on the original Transformer architecture(base model) for Russian Language. The authors believe that this task should be categorized as a sequence-to-sequence classification problem. The dataset used in this paper is the SynTagRus dataset. The authors show two transformer based models for POS tagging (Base model and the simplified model which is an improved form of the base model). This paper concluded that the base model gave a micro-averaged F1-score of 0.970 and the simplified model gave one of 0.974. In order to implement the POS tagging the intention is to use a similar simplified model.

Popular word embedding models are introduced by the authors of [3] and desired properties of word embedding models and their evaluators are discussed. The evaluators are categorized into two types, intrinsic and extrinsic, and their experimental results on six word embedding models are evaluated. Finally, they adopted a correlation analysis to study performance consistency of extrinsic and intrinsic evaluators. The authors provide us with valuable guidance in selecting suitable evaluation methods for different application tasks. This paper gives different ways in which the model can be evaluated.

The authors of [4] propose a simple architecture which is the Transformer, which is solely based on attention mechanisms, with convolutions and recurrence entirely. Experiments that were conducted on two types of machine translation tasks showed that these models perform better and are more parallelizable, hence it required less training time. The authors' model achieved a BLEU score of 28.4 on the 2014 WMT English-to-German translation task, this improved over the existing benchmark, by over a score of 2 BLEU. On the 2014 WMT English-to-French translation task, their model established a BLEU score of 41.8. This paper provides the architecture of the transformer model which is to be implemented.

III. Proposed Solution

A dataset must initially be identified which will be used to train the model. From the nltk library, three datasets have been selected and merged, the brown corpus, the treebank corpus and the conll2000 corpus. The brown corpus consists of 57340 tagged sentences, the treebank corpus consists of 3914 tagged sentences and the conll2000 corpus consists of 10948 tagged sentences. Together these make up a total of 72202 tagged sentences. Out of these 57761 tagged sentences are used as the training data, 7220 for the testing data and 7221 as the validation data. Every word in the three corpora takes on one of thirteen possible tags.

Each word along with its tag in each sentence in each corpora is represented as a tuple. These need to be appropriately preprocessed to convert them into an appropriate form that can be fed into the BERT model. First each sentence in the form of a word-tag pair in the final corpus needs to be split into two sets of data, one of them is a list of sentences, the other is a list of tags for each sentence. BERT requires special tokens to understand its input sequence properly, the "[CLS]" tag at the beginning of the input sequence is a special classification token and the last hidden state of BERT corresponding to this token is used for classification tasks. The "[SEP]" tag at the end of the input sequence allows BERT to understand the separation between two input sequences. To account for these changes, the POS tags must have special tokens at the beginning and end as well, known as padding and given by "-PAD-".

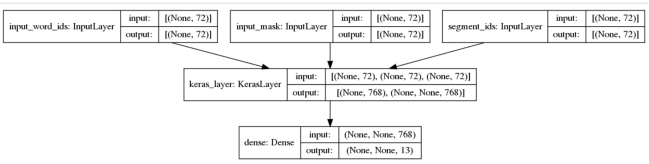
The data consisting of the tags is then mapped such that each tag is associated with a numerical value. The textual data is then fed to the BERT preprocessor model where each word is replaced by a contextless unique ID to that word. These numerical forms are more robust to conventional deep learning techniques where a considerable amount of mathematical operations need to be performed on the input to get a valid output. This also ensures that each input is of a fixed length, which makes matrix related operations much simpler to perform. The tags are then converted into a one-hot encoded form as this task is one of multiclass classification.

The deep neural network is then constructed. The inputs to the model are the outputs obtained from the BERT preprocessor model. These consist of the word IDs, input type IDs and the input masks. The first part of the model is the BERT architecture. The version of the BERT architecture being used has twelve hidden layers, 512 neurons per hidden layer and twelve attention heads. The outputs from the BERT layers consist of the contextual word

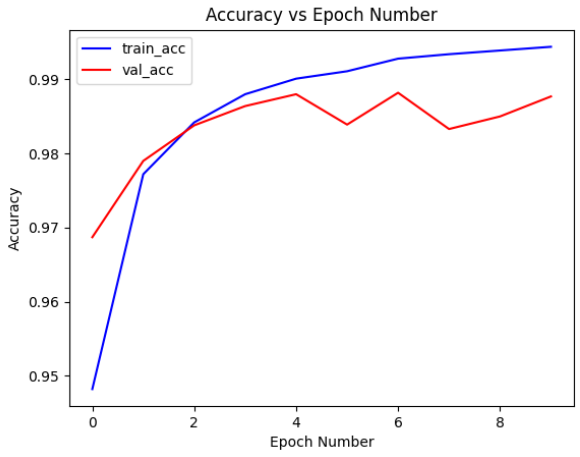
embeddings of the input sequence, each output embedding being a vector of size 768. These word embeddings are then fed to a dense output layer using the softmax function in order to classify each word embedding in the sequence.

The summary and diagrammatic representation of the model can be seen below.

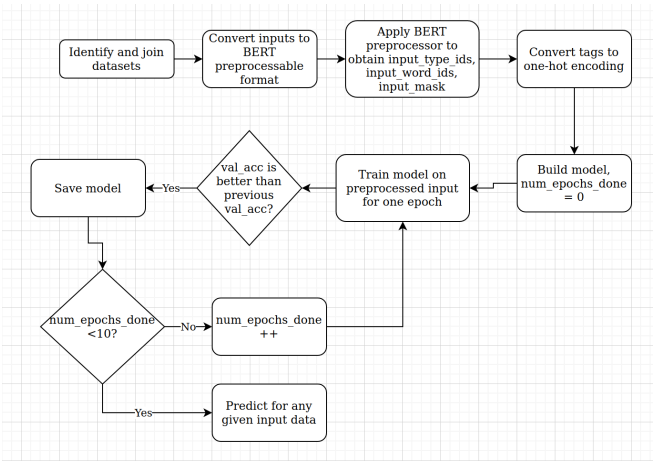
```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_word_ids (InputLayer)     [(None, 72)]         0
__________________________________________________________________________________________________
input_mask (InputLayer)         [(None, 72)]         0
__________________________________________________________________________________________________
segment_ids (InputLayer)        [(None, 72)]         0
__________________________________________________________________________________________________
keras_layer (KerasLayer)        [(None, 768), (None, 768)]  109482241   input_word_ids[0][0]
                                                                 input_mask[0][0]
                                                                 segment_ids[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, None, 13)     9997        keras_layer[0][1]
==================================================================================================
Total params: 109,492,238
Trainable params: 109,492,237
Non-trainable params: 1
__________________________________________________________________________________________________
```



The plots for training accuracy against epoch number and validation accuracy against epoch number can be seen below.



IV. Flowchart/Algorithm



V. Experimentation

Given any random sentence or sequence of sentences, the model is capable of generating POS tags for each word in the input. A simple illustration of this is shown below.

As this above sentence is a very basic one, the model was then tested on 7221 sentences from the NLTK library which it was not trained on. These 7221 sentences were randomly selected as validation data from a combination of the three datasets used before the model had been trained. The classification report generated from the testing of these sentences can be seen below.

	precision	recall	f1-score	support
0	0.00	1.00	0.00	0
1	0.94	0.93	0.94	10899
2	0.97	0.97	0.97	18427
3	0.95	0.94	0.95	6948
4	0.98	0.97	0.97	4781
5	0.98	0.98	0.98	17130
6	0.96	0.97	0.97	38589
7	0.96	0.96	0.96	2813
8	0.97	0.97	0.97	6043
9	0.94	0.93	0.93	4176
10	0.98	0.97	0.97	19251
11	0.98	0.97	0.97	23739
12	0.92	0.89	0.91	752
accuracy			0.96	153548
macro avg	0.89	0.96	0.88	153548
weighted avg	0.97	0.96	0.97	153548

To understand what each class represents, the mappings for each class are as follows.

```
{0: '-PAD-', 1: 'ADJ', 2: 'ADP', 3: 'ADV', 4: 'CONJ', 5: 'DET', 6: 'NOUN',
 7: 'NUM', 8: 'PRON', 9: 'PRT', 10: 'PUNCT', 11: 'VERB', 12: 'X'}
```

The f1-score of class “-PAD-” is 0.00 as every embedding beyond the count of seventy words is classified as zero, this skews the report and all classifications of zero have hence been omitted. The class “X” has a relatively lower f1-score as the number of training instances with the class “X” is very low, only 752 instances. The overall accuracy of the model is 0.96 and the weighted average of the f1-score is 0.97.

As the model in this paper has been trained on datasets present in the NLTK library, a comparative study between the outputs of the NLTK library’s POS tagger and this model’s POS tagger has also been made. To perform a simple analysis, fifty random sentences have been generated and fed to both NLTK’s POS tagger as well as the model in this paper.

For each sentence that is fed, it can be seen that there are differences in the two predictions and in multiple cases the BERT based model’s predictions are more accurate than those of NLTK’s POS tagger. Consider the sentence “We're careful about orange ping pong balls because people might think they're fruit.” As can be seen below, NLTK’s POS tagger predicts the word “orange” to be a noun whereas the BERT based tagger predicts it to be an adjective. In this case the BERT based tagger has made the correct prediction.

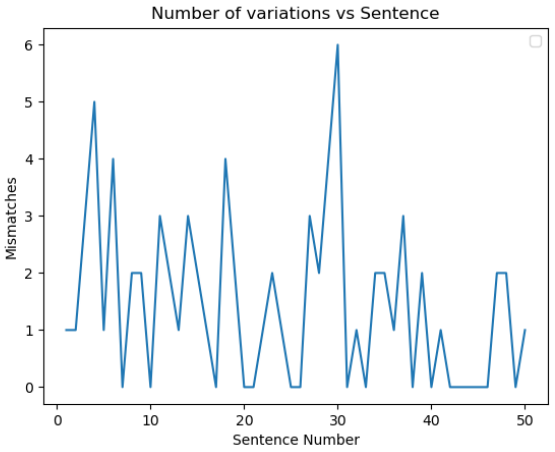
Word	NLTK Tag	BERT Tag
we	Pronoun	Pronoun
are	Verb	Verb
careful	Adjective	Adjective
about	Adposition	Adposition
orange	Noun	Adjective
ping	Noun	Noun
pong	Noun	Noun
balls	Noun	Noun
because	Adposition	Adposition
people	Noun	Noun
might	Verb	Verb
think	Verb	Verb
they	Pronoun	Pronoun
are	Verb	Verb
fruit	Adjective	Noun
.	Puncutation	Puncutation

Another example to consider is the sentence “Make sure you tip-toe downstairs.” As seen below, the word “make” is predicted as a noun by NLTK whereas it is in fact a verb as predicted by the BERT based model.

Input Sentence: Make sure you tip-toe downstairs.

Word	NLTK Tag	BERT Tag
Make	Noun	Verb
sure	Adjective	Adjective
you	Pronoun	Pronoun
tip-toe	Adjective	Verb
downstairs	Noun	Adverb
.	Puncutation	Puncutation

A graph plotting the number of discrepancies in the two predictions for each sentence has been plotted below.



VI. Conclusion

The model in this paper achieved a validation accuracy 0.96 and an f1-score of 0.97 on a subset of the combination of three datasets, the brown corpus, the treebank corpus and the conll2000 corpus. For any given sentence, when the output of the BERT based model is compared with the output of NLTK’s POS tagger, there are on average two discrepancies. More often than not it can be shown that the BERT based model predicts it more accurately than NLTK’s POS tagger.

VII. Future Works

The model used in this paper has been trained on a combination of datasets present in the NLTK library. Training could be tried on a benchmark dataset such as the Penn Treebank dataset. Since the number of unknown tagged words were minimal in the dataset, adding them would give a better model.

POS tagging is useful in several applications such as Named Entity Recognition, question answering, sentiment analysis, etc. The intention is to use POS tagging to generate images given a sentence or a phrase. The application would also be an extension of the use of the transformer architecture.

References

[1] Liu, Jerry, Nathan O’Hara, Alexander Rubin, Rachel Draelos, and Cynthia Rudin. "Metaphor detection using contextual word embeddings from transformers." In *Proceedings of the Second*

Workshop on Figurative Language Processing, pp. 250-255. 2020.

[2] Maksutov, Artem A., Vladimir I. Zamyatovskiy, Viacheslav O. Morozov, and Sviatoslav O. Dmitriev. "The Transformer Neural Network Architecture for Part-of-Speech Tagging." In *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pp. 536-540. IEEE, 2021.

[3] Wang, Bin, Angela Wang, Fenxiao Chen, Yuncheng Wang, and C-C. Jay Kuo. "Evaluating word embedding models: Methods and experimental results." *APSIPA transactions on signal and information processing* 8 (2019).

[4] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Advances in neural information processing systems*, pp. 5998-6008. 2017.

[5] https://jalammar.github.io/illustrated-bert/