Bangla POS Tagging With No Training Data

Omar Faruqe
Dept.Electrical & Computer Engineering
North South University
omar.faruqe15@northsouth.edu

Md Tasbiul Hasan
Dept.Electrical & Computer Engineering
North South University
tasbiul.hasan@northsouth.edu

Dr. Nabeel Mohammed
PhD. Associate Professor
Dept. Electical & Computer Engineering
North South University
nabeel.mohammed@northsouth.edu

***Abstract* - Our vision is simple to create Bangla POS tagging with no training data without no handicraft resources and without any languistics information. And our model is English baseline though recently used many BNLP tools for POS tagging but we are focusing on English model.**

Index Terms- POS tagging, LSTM, BERT, Word Embedding

I. Introduction

In any Natural Language Processing (NLP) pipeline, POS tagging is one of the baseline tasks like language modeling, sentiment analysis, and named entity recognition.
POS Tagging indicates the part of speech for a word; annotating each word in a sentence gives an appropriate tag on that particular word. The performance of POS tagging is very high for several languages, like English, German, Italian and Chinese. Systems using supervised machine learning techniques based on manually annotated datasets predefined tags, for Sequence tagging widely used UDPOS dataset where train data 12543, valid data 2002 and test data 2077. But in south Asian countries, there is no standard pos tagger and no predefined tag dataset. In our research-based project, we analyze Bangla pos tagging without any training data based on the English model. We used the word and character level distributed representation with LSTM neural networks. The motivation of using such an approach is its success rate in a sequential task. We also try fine-tuning on BERT model because of better performance.

II. Literature Review

Part of speech tagging is a simple but useful form of linguistic analysis. Parts of speech tagging means grammatical feature labeling in a sentence. It is similar to tokenization for coding languages. Assigning a part of speech tag for each word in handwriting is very time-consuming. For getting jobs done with various automated techniques apply. Automated POS tagging is a technique to automate the lexical categories' annotation process. The process takes a word or a sentence as input, assigns a POS tag to the word or each word in the sentence, and produces the tagged text prediction as output. But the challenging task is a proper prediction on POS tagging. Many NLP researchers are working for that. POS tagging techniques have been implemented for English and many other Western languages and show a satisfying performance of near than 97%. Several methodologies have been utilized for that. The most eminent ones are principle-based, stochastic, or change-based learning techniques. Principle-based taggers assign a tag to every word utilizing an arrangement of manually written rules. These guidelines could determine that a word after a determiner and a modifier must be a thing for an occurrence. After that, the arrangement of principles must be appropriately composed and checked by human specialists. The stochastic approach corpus gives a most biased tag for a specific word. Different probabilistic methodologies apply for POS labeling. In a stochastic approach, issues arise about ambiguity. The corpus gives a most biased tag for a specific word based on the number of frequency.A word can be used in different roles in different sentences. The change-based methodology joins the standard-based methodology and factual methodology. In most cases, English and many western languages are implemented by many POS tagging techniques and show outstanding results. But in south Asian languages like Bangla, Hindi pos tagging resources are very rare. Previously work on Bangla pos tagging has been reported by Chowdhury et al. (2004) and Seddiqui et al. (2003). implemented a rule-based POS tagger, which requires handcrafted rules by human experts and many years of effort from linguists. But not show performance analysis of their work, the feasibility of the proposed rule-based method for Bangla is suspect.No review or comparison of Bangla Pos tagging was available in that paper. They focused on more morphological analyzers than a POS tagger. The International Institute of Information Technology (IIIT), Hyderabad, India, initiated a POS tagging contest, NLPAI ML-Contest061, for the Indian languages in 2006. Several

teams came up with various approaches, and the highest accuracies were 82.22% for Hindi, 84.34% for Bengali and 81.59% for Telugu. The SPSAL2007 workshop in IJCAI-07, a competition on POS tagging and chunking for south Asian languages, was conducted by IIIT, Hyderabad. The best POS tagging accuracies reported were 78.66% for Hindi (Karthik, 2007),77.37% for Telugu (Karthik, 2007), and 77.61% for Bengali (Dandapat, 2007). The POS taggers for Bengali can be found in Ekbal et al. (2008) with ME, Ekbal et al. (2007) with CRF, and Ekbal and Bandyopadhyay (2008a) with SVM based approach. Our approach is to build a Bangla pos tagging using the English model baseline. The motivation of our approach Bangla is our mother tongue, but we have no standard POS tagger. Focus on implementation lack of resources. Existing resources are not available and difficult to develop. We are not using any handcrafted features. For our experiment, we are using the bi-directional LSTMs - CRFs, in which we exploited word- and character-level distributed representations.LSTMs are a variant of RNN due to the power of capturing long-term dependencies. Many NLP tasks like Language modeling, sequential task labeling used RNN. RNN maintains previous information using memory cells and predicts the output.For example, it takes a sequence of vectors (x1,x2,x3,...,xn) as input and produces sequence (y1,y2,y3,....,yn) as output.Issues arise on RNN can not capture information from a very long distance because of gradient vanishing problem. Which results that RNN gives biased results based on the most recent input. This is where LSTMs came in to overcome the RNNs issues and at the same time capture long-term dependencies using gating mechanisms. The reason for using bidirectional LSTMs in our experiment, forward LSTMs can capture suffix information with their final state, whereas the final state of the backward LSTMs can represent the prefix information.

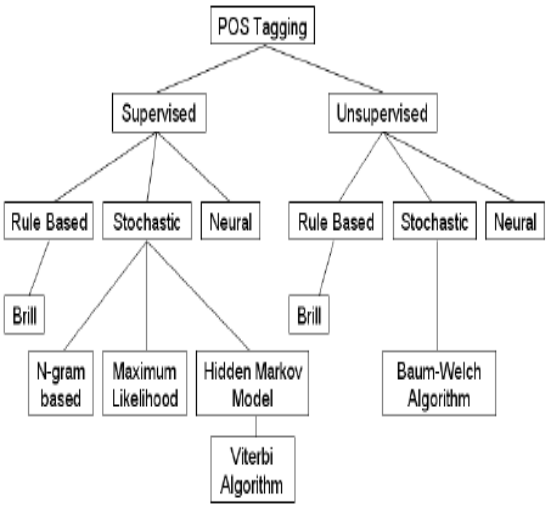

Figure 1: Different POS tagging Model

III. Proposed Methodology

In our Experiment, we used the LSTM model, which is the RNNs variant. We present a single LSTMmemory cell for clarity. It uses several gates, such as input, output, and forget, which control the amount of information to get- in and out to the LSTMcell. Gates are designed using sigmoid neural net layers and pointwise multiplication operators. This version of LSTM does not use “peephole” 1 connections. The LSTM memory cell is implemented using the following equations

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

Here, Where sigma is the point-wise logistic function and dot is the point wise multiplication of the two vectors i,f,o, and c are the input gate for gate, output gate, and memory cell vectors. The weight matrix represents the weight vectors of different gates. For example, Wx is the input gate matrix bi, bf, bc, and bo denote bias vectors. For a sentence with words with input (x1, x2,... xn), the LSTM will compute an output representation of (h1, h2,... hn).

We also used Fine-tuning on the Bert model. Though, we have no deep knowledge on BERT. We use a reference model based on that model. We take some sentences and convert that sentence and word with google translator and show the performance. The motivation of the BERT model added in our project is that its capable memory capacity is better than LSTM architecture, requires less data, and is less biased than the LSTM model.

IV. EXPERIMENTAL SETUP

A. Dataset

Though we are not concerned about the dataset. We kept our architecture normal; we worked with some sequential sentences. First, we try to use Prothom-alo (2013) Bangla datasets. Try to annotate data due to a lack of data annotated knowledge. We try to take only Bangla content in a text file with sequential order and ensure that one sentence per line is like this:

বরিশাল সেপ্টেম্বর ১২ বিডিনিউজ টোয়েন্টিফোর ডটকম
উচ্চশব্দে গান বাজানোর প্রতিবাদ করায়

নরসিংদীতে একজন খুনের অভিযোগের পরদিন বরিশালে এ ভাবে গান বাজানো নিষিদ্ধ করেছে

With google translator, we also translate our Bangla dataset into English. Here, we faced a problem. We use only Google translator, but here, the text size is fixed, and all Bangla words are not turned into English. Then we used google translate API, which easily converted our dataset into English. Also, ensure that one sentence per line.

B. Evaluation Metrics

If the project had been successful, the evaluation would be straight forward.1)Use language translator to convert Bangla to English 2)take some sequence as inputs in English model 3) use softmax to predict some value which is our ground truth value. 4)then, take some predicted values based on the same model as a Bangla sentence. Predicted values give us better results, or bad results are not concerning issues. Also, compare English and Bangla accuracy. But due to lack of skill, we can not implement the 3rd and 4th stages properly. We have not been able to complete the training phase properly. We initialize our ground-truth value in the training phase. Also, our selected approach is poor in our proposed Bangla Pos tagging with no training data.

B. Model

In our experiment, we used LSTM sequence models. Sequence models are central to natural language processing, where there is some dependence through time between our inputs. In the case of an LSTM, for each element in the sequence, there is a corresponding hidden state to predict words in a language model, part-of-speech tags, and a myriad of other things.

The semantics of the axes of these tensor is important. The first axis is the sequence itself, the second index in the mini-batch, and the third indexes elements of the input. We ignored mini-batching and assumed it always has just one dimension. In our LSTM model, we took some Bangla to English sentences. We have embedding characters. The character embeddings will be the input to the character LSTM.
The model is as follows, let our input sentence be W1,....., Wm, where W is our vocabulary. Also, Let T be our tag set and y the tag word w.Our prediction
of the tag of word W by y.This is our structure prediction and model where our output is a sequence y1,.....,ym.

For prediction, pass an LSTM over the sentence and the hidden state at time step i as h. We assigned each tag a unique index. In our project, we had word_to-ix in the word embeddings coding section. Then our prediction purpose, we used the log softmax of the affine map of the hidden state, and the predicted tag is the tag that has a maximum value. Also, try to fit the same process in English to Bangla translate sentences in the LSTM model and get the prediction.

We also used fine-tuning on the Bert model. Though, we have no better knowledge of the BERT model. For better prediction, we try to fit some sentences on Bert Model. My reference model predicted the English pos tag. We tried to implement with two functions: Banga text to convert English, and another is an English word to convert Bangla used in google translator API.

V. Result

Our model is not supposed to be good. The words it can predict accurately are mostly words which the model had encountered during training and remembers during test time. Whenever it encounters a new word, it fails badly. We implement our model from the primary stage. We have no deep knowledge of LSTM architecture. And fine-tuning on the BERT model gives better accuracy, like 99.91%. We only implement with a predefined dataset available in PyTorch torch text and other resources. we convert sentences and words on the reference model and show the result.

VI. Conclusions

In our paper, we try to present a neural network-based POS tagging for Bangla, which doesn't require any hand-crafted features or knowledge sources in linguistics information. Try to focus on building a Bangla pos tagging model and performance based on simple LSTM architecture. In our model, fine-tuning on Bert gives better performance, and it has a more memorized capacity and requires fewer data. Further, this model can be the solution for the sequential model.

References

[1] https://www.researchgate.net/publication/311871941_Bidirectional_LSTMs_-_CRFs_networks_for_bangla_POS_tagging

[2] https://www.cs.ubc.ca/~carenini/TEACHING/CPSC503-09/FINAL-REPORTS-08/hammad-report1.1.pdf?fbclid=IwAR0GTVVUusesU8o-HQCapmw2nblNKhlX1eU4OmxnHXIJF17gh_WTwDi4bRs