Report

BIS Token Classification

Refinements on the project

Author: Mahammad Namazov

Date: 2023-01-11

Abstract

The project is an approach to classify tokens in the given sentences into three classes which are Beginning (B), Intermediate (I) and Space (S), so that it is called BIS Token Classification. I implemented Bidirectional Long Short Term Memory (Bi-LSTM) model as a base model along with Linear Neural Network model with one hidden layer and one output layer. On the other hand, I optimal data processing model to read and prepare the raw data for the model, in order to minimize memory usage. After several experiments, F1 score 0.8002 was obtained.

Keywords: BIS, Bi-LSTM, Neural Networks, Deep Learning, Data Processing

Contents

1. Introduction

Token classification is one of the important tasks in Natural Language Processing [1] field of Artificial Intelligence, which can be used as character tokenizer tool in further projects. The aim of this very project is to classify tokens according to their positions in the words of provided sentences, which characters can be classified as Beginning token (B), Intermediate token (I) (i.e., it stands either in the middle of the word or it is the final character) and Space (S).
Project was a bonus project during my Natural Language Processing (NLP) course, so that I had access to the dataset. However, because of ambiguities in confidentiality, I will not be able to provide the dataset for this task. In this report I will introduce my work and approach and further prospective works, along with results that I have obtained.

2. Background

It is fact that, end-to-end neural models are trained to learn specific patterns among provided data. According to the process that data is involved, learning loss is tried to be minimized by using Deep Learning models. In this kind of project (BIS Token Classification), the main approach could be processing data in a way that pattern matching might increase training in addition to testing several Neural Network Models. That rounds my main approach up on the given problem. In order to evaluate the model, F1-score, accuracy computation on correct predictions over development dataset and confusion matrix over three labels were computed.

3. Data

Dataset includes raw sentences and their ground truth values as a sequence of corresponding B, I and S labels. The main structure of data and character based labels can be seen from the examples which is expressed in Table 1. Notice that, examples are provided in the table are not part of dataset. On the other hand, it is important

Sentence	**Ground Truth**
I am looking forward to hearing from you!	BSBISBIIIIIISBISBIIIIIISBIIISBIII
Artificial Intelligence is definitely fun.	BIIIIIIIIISBIIIIIIIIIIISBISBIIIIIIIIISBIII
I love Mathematics.	BSBIIISBIIIIIIIII

Table 1: Several examples to see the data structure

to run some statistics over dataset so that we can have some insights about possible errors that we are not expected. For instance, our task is character classification and we know that we will be busy with some letters, numbers and punctuation. However,

using this assumption and building dataset only on english language characters, numbers and punctuations can mislead not only us but also our model. So that vocabulary of the task must be built dynamically and some numerical statistics must be run about data distribution 2. It is seen from the table that I is the most annotated label in the dataset. We will use this information for majority baseline F1 score [3] to compare our model.

Dataset	**B**(eginning)	**I**(ntermediate)	**S**(pace)
Training	3	2	1
Validation	3	d	3

Table 2: Label distribution over datasets

3.1 Data preparation

In this section I will explain processing procedure of the data. In NLP related task, it is significant to build the vocabulary, which includes information about the raw data and their indexed version. Since our task is character based classification, vocabulary elements are characters and their indexes. Notice that vocabulary is built over train dataset, since validation dataset must be unknown to the model during the training phase and model must be ready for out-of-vocabulary (OOV) words. Additionally, making model case-sensitive can also affect the results, so that each task must have separate vocabularies. While vocabulary has 343 elements in case sensitive approach, it includes 121 elements in approach where all characters are lower-case.
In order to make model learn patterns I used window technique. Firstly, sentence is transformed into list of characters (or tokens, where token is used for characters rather than for words). When window size is n, which specifies number of tokens can be involved by window), and window shift is m, which specifies step size that window will be moved over the given sentence), I will be able to represent one list of sentence in the form of several windows where pattern among n tokens can be taught to the model. Notice that, choice of smaller m and n can bring more specific patterns and increase amount of data combinations to be analyzed by the model. In case the value of window shift is chosen as 1, then maximum number of windows is equal to number of tokens in the sentence.
In such scenario, padding technique is very useful, since it helps to keep all windows in the same size, where we add special token for padding element into our vocabulary. It must be emphasized that in loss and accuracy computation, padding element is ignored since it is redundant for original dataset and model improvement.

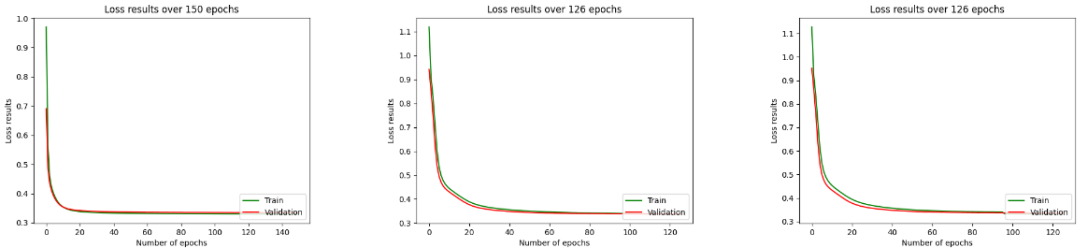
4. Approach

I used Bi-LSTM [2] model as a base model of the classifier and I added Fully Connected Network (FCN) to the top of this model to classify tokens into the given labels. I used Categorical Cross Entropy Loss as a loss function and SGD as an optimizer with

dynamic learning rate. Bi-LSTM model was fed with the embedding data which is in size of vocabulary size x embedding dimension, where embedding dimension is 300. Hidden dimension of the Bi-LSTM model was chosen 150 and number of LSTM layers is 2. Dropout technique was used to eliminate the overfitting phenomena. Although, I have done several experiments, I will share three of them which outperform other experimental results.

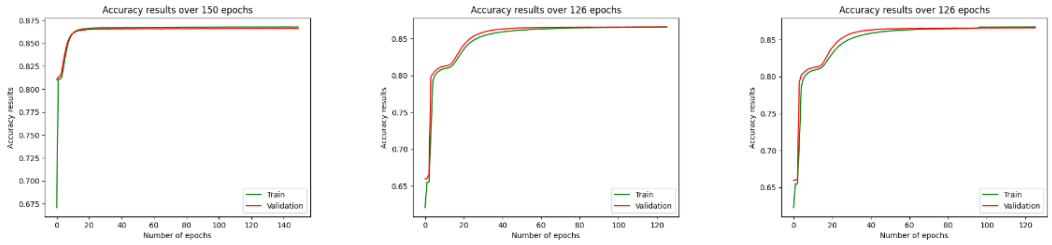
5. Results

In this section I will share training and validation graph results along with confusion matrix evaluation at the end of the training. Additionally, I will share baseline F1-score which was computed according to the major label in the train dataset. Models were trained over 100 epochs, with window size 100 and window shift 100 in order to feed model with longer patterns and increase training time. However, this values can be used differently in further experiments. Figure **??** and Figure **??** represent accuracy and loss results, respectively during the training and validation phases. It can be seen that last two models were trained 126 epochs but the first one was trained for 150 epochs.



(a) (b) (c)

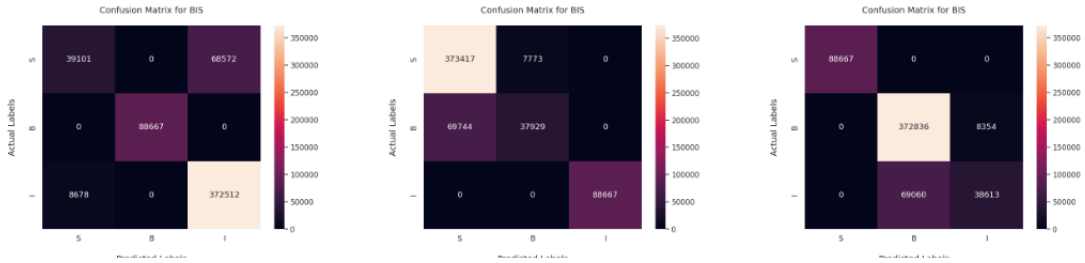
Figure 1: Loss computations over validation data for the first (a), second (b) and third (c) approaches



(a) (b) (c)

Figure 2: Accuracy computations over validation data for the first (a), second (b) and third (c) approaches

Additionally Figure 3 represents confusion matrix over validation data for each approach. In order to use more specific information, we enumerated models which



(a) (b) (c)

Figure 3: Confusion Matrix over validation data for the first (a), second (b) and third (c) approaches

Parameters	Model 1	Model 2	Model 3
Bidirectional	True	True	True
# LSTM layers	1	2	2
epoch change	115	100	100
learning rate	0.0003 - 0.001*	0.0003 - 0.001*	0.0003 - 0.001*
dropout linear	False	True	True - False **
dropout LSTM	False	True	True - False **
# linear layers (hidden + out)	2	2	2

Table 3: Model Parameter changes.
* Learning rate was modified from the first value to the second after the epoch which is given in epoch change row of the table
** Dropout was removed after the epoch which was given in epoch change

parameters and changes can be found in Table 3.

Additionally, baseline F1 score, which was computed over the label with the highest frequency, is same for every experiment that is 0.16, since window size and window shift values did not change during experiments. However average F1 score with macro average that we obtained over three experiments is 0.80. We provide average value, because of very small changes in each experiment results.

6. Discussion

It can be seen from the loss and accuracy graphs that there is not any overfitting during the learning process. However, confusion matrix shows that each model is confident in prediction of specific labels. For instance the first model predicts correctly label B in every opportunity, although it predicts the other labels in several cases, incorrectly. The same phenomena occurs for Model 2 and Model 3 for I and S labels, respectively. Following further refinements can be made:

- Model can be tested with different window size and window shift values

- Complex or simpler models also can be made and process can be repeated
- Pre-trained embeddings can provide better results

Bibliography

[1] Hannes Hapke, Cole Howard, and Hobson Lane. *Natural Language Processing in Action: Understanding, analyzing, and generating text with Python*. Simon and Schuster, 2019.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[3] Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*, pages 1015–1021. Springer, 2006.