Report and Code Running Instructions

Avijit Roy, IIT Kanpur

Abstract

In this document, the method to get the accuracy and other instructions are provided. The problem of fact-checking is solved here using truthfulness classification and deep learning. We use LSTM to model the text.

Method Description

First we need to clean the dataset. We remove all the missing data columns and remove all the special characters from the data. Then we save it as another TSV file. For binary classification we combine true, mostly_true and half_true as one class and false, barely-true and pants-fire as another class.

To classify the given text we used Long Short Term Memory(LSTM) networks. I used PyTorch's own embedding to map the words to a 300 dimensional vector. Then those vectors are passed to an LSTM to get a representation of sentence. Then by passing that representation through a linear layer and softmax layer we get the probability of each sentence to be in some category.

The dataset is concatenated and passed to the network for prediction. For that we concatenate justification, statement, job_title and party. The sequence of concatenation is important.

We train three of these networks with different initialization and then ensemble them to get the final prediction.

We use the same network structure for binary classification as well.

Different Ideas Tried

- I cleaned the data and removed missing value columns
- I tried different models like adding another linear layer with ReLU activation function. But it started overfitting.
- I built another model where the statement and justification are passed to two parallel LSTM and later their vectors are concatenated and passed to a linear layer. But this model did not increase the accuracy.
- I tried concatenating the inputs in different orders and then passing it to the network and it actually gave different results. When I concatenated justification before statement it gave better results then concatenating after.
- I tried an ensemble of three models trained separately. And the accuracy increased.

Accuracy

For Six-Way Classification:

- TEST: 26.07 percent
- VALIDATION: 26.71 percent

For Binary Classification:

- TEST: 64.10 percent
- VALIDATION: 61.97 percent

Instructions to Run the Code

CODE CAN ONLY BE RUN IN AN GPU ENABLED ENVIRONMENT

Required Libraries:

1. PyTorch 1.1.0
2. sklearn 0.20.1
3. pandas 0.24.2
4. numpy 1.13.3

Now download the Dataset from: https://github.com/Tariq60/LIAR-PLUS/tree/master/dataset and keep the files test2.tsv, train2.tsv and val2.tsv inside dataset folder

Now run these commands from root folder:

1. python load_dataset.py

FOR SIX WAY CLASSIFICATION:

1. python network_1.py
2. python network_2.py
3. python network_3.py (These can be run parallelly) (FOR TRAINING THE NETWORK)
4. python test.py (FOR TESTING)

FOR BINARY CLASSIFICATION:

1. python network_binary_1.py
2. python network_binary_2.py
3. python network_binary_3.py (These can be run parallelly) (FOR TRAINING THE NETWORK)
4. python test_binary.py (FOR TESTING)

THE SAME CAN BE FOUND AT readme.txt

References

1. Where is your Evidence: Improving Fact-checking by Justification Modeling Tariq Alhindi , Savvas Petridis, Smaranda Muresan
2. Pytorch, sklearn and Pandas
3. https://pytorch.org/
4. https://github.com/clairett/pytorch-sentiment-classification