Amarandei Alexandru, Mosor Andrei, Pelin Ioana, Smău Adrian

Team-Leader: *Cușmuliuc Ciprian Gabriel*

April 2021

CLEF2021 - CheckThat! Lab

Week 3 Report

Introduction

This team was assigned with the task of implementing an algorithm for detecting Fake News in Python. During the last week, the team's work was focused on improving performance on the current solutions and implementing a BERT approach. During this week's run, one of the two sub-teams is focusing on adding two new tools to the arsenal, whilst the other one is improving on the current ones.

Methodology

Overview

For the remainder of the document the first sub-team will be referred to as the 3Layer Team and the second one as the BERT Team.

The 3Layer Team's main purpose was to experiment with POS (Part-of-Speech) Tagging and Semantic Analysis.

The main driver behind including POS Tagging is that numerous statistics have shown that there are certain correlations between the frequency of some parts of speech and the veracity of the facts spoken in the respective sentence (e.g. too many adjectives describing the same noun are usually raising a flag). As for the Semantic Analysis, certain topics (e.g. politics, armed attacks) tend to be more prone to appearing in fake news than others, thus that relation should be further explored.

The BERT Team's main purpose was to get familiar with the BERT technology, while simultaneously designing a performant and accurate model, able to successfully train and predict based on any given text. This week, the team has mainly focused on researching pre-processing techniques, training methods, optimisation, fine-tuning, tokenizing, mapping, masking, tensors etc., all essential components of an efficient BERT-based algorithm.

3Layer Team

Data Preparation

The Data Preparation stage consisted of:

- removing the **public_id** column from the CSV files received
- combining the two datasets
- combining the **title** and **text** columns
- removal of punctuation signs
- removal of stop-words
- removal of dashes and underscores
- lowercasing the text
- lemmatization of text

TFIDF Vectorization

This step was applied on the newly-created **clean_text** column, using a bigram (a contiguous sequence of *n* items, where *n* is 2).

POS Tagging

For this step, the team used the spaCy NLP (Natural Language Processing) library on the **text** column to obtain the original text in POS form, then saving it in the **POS_text** column.

Semantic Analysis

The last approach of the 3Layer Team was a semantic analysis using Stanford's Empath Tool to categorise the words in the tweets by their lexicon, this operation being performed, again, on the original text. The resulted form was stored in the **semantics_text** column.

Models

After the afore-mentioned operations were made, a TFIDF vectorizer was used on each of the newly-created columns and the following classifiers were used: NB (Naive-Bayes), RF (Random Forest) and GB (Gradient Boosting).

The results were good for each of the individual classifiers on each of the representations and will be shown in the **Results** section and elaborated on in the **Discussion** section below.

However, the best performance was recorded after using the three representations together, each having a different weight (hence the name 3Layer). Both the Accuracy and the Macro Average have increased, the best ration being on the GB Classifier.

BERT Team

Preprocessing

- the text must be cleaned, so that the tokenization is optimal, only letting alpha-numerical characters in the text column
- integration of the title and text in the same column (as both are of relevance when trying to label a text)
- removing stop words and lemmatization

Data Exploration

Data exploration is a mandatory step in the implementation, as it is necessary to grasp the type and structure of the data at hand.

Modelling - Tokenization and Parameter Preparation

- the data must be thoroughly modelled in order to be eligible
- the tokens must be prepended with the [CLS] Token and appended with the [SEP] token
- the sentences must be truncated or padded according to the chosen max length
- **input_ids** and **attention_masks** must be generated in order to later create the TensorDataset

Fine-Tuning

Trial and error process of finding the optimal batch size, number of epochs and learning rate.

Training and Validation

A 90/10 train-to-test ratio was used.

Results

3Layer

TFIDF Vectorization on Cleaned Text

Classifier	Accuracy	Macro Average
Multinomial Naive-Bayes	0.74	0.32
Random Forest	0.71	0.22
Gradient Boosting	0.62	0.23

Note: Although the accuracy was high, that was due to the disproportionate number of False texts.

TFIDF Vectorization on POS Tags

Classifier	Accuracy	Macro Average
Multinomial Naive-Bayes	0.71	0.21
Random Forest	0.72	0.26
Gradient Boosting	0.69	0.35

Note: As expected, although the Accuracy is still high, the Macro Average is still on the low side.

TFIDF Vectorization on Semantic Tags

Classifier	Accuracy	Macro Average
Multinomial Naive-Bayes	0.71	0.21
Random Forest	0.72	0.28
Gradient Boosting	0.62	0.25

Note: The results are certainly close, which means that there are some correlations between all three of the representations.

TFIDF Vectorization on All Three Representations, using a sparse matrix form

Classifier	Accuracy	Macro Average
Multinomial Naive-Bayes	0.75	0.32
Random Forest	0.76	0.26
Gradient Boosting	0.77	0.33

Note: The Accuracies are the highest so far and the mean of the Macro Averages has also increased.

BERT Team

Other than finalising research and solving out all errors before the final implementation in Week 4, there were no tangible results during this week's run.

Discussion

3Layer

The combination of the three representations yielded very promising results with each of the classifiers (used with the default parameters) and has now become the main focus of this sub-team.

By degree of objectivity, the percentages (i.e. weights) used for the representations were and will remain from now on:

Weights

Representation	Weight
Clean Text TFIDF	0.5
POS Tagging	0.15
Semantic Analysis	0.35

Note: Multiple distributions were tried, however, the one above performed the best amongst all.

The great discrepancy between the False texts and the other three categories of texts (True, Partially False and Other) resulted in a very good classification of False texts and a very poor one of the rest.

Although artificially adjusting the label distribution is on the table, it would certainly be more beneficial to have a more balanced dataset.

BERT Team

During this research week, the team never came across a 4-way algorithm, which is important since the dataset has two more labels other than *True* and *False* (i.e. *Partially False* and *Other*).

The team was in need of finding a way to get the BERT Classifier to make a 4-way prediction.

The main problem that the team has faced during this week was continuously having to deal with the '*CUDA error device side assert triggered*' error when running tests.

Another point of discussion is the great number of *False* labeled entries in the dataset. The team had to find a way of preventing overfitting (an approach involved making all entries equal in numbers with *Other* labeled entries, as they represented the smallest portion of the dataset, but that resulted in roughly 40 entries per label, which meant that the algorithm would be under-trained).

Lastly, the resources provided by the *Google Colaboratory* platform were insufficient, as the program frequently crashed due to low RAM availability.

Future Approaches

3Layer

Hyper-Parameter Tuning

As the last week's run has shown, hyper-parameter tuning can certainly improve results and will be used on all classifiers from now on to further increase the Macro Average.

Models

The Classifiers used during this week will remain in the toolset, however, a KNN (K-Nearest Neighbours) Classifier will also be included in future runs as it yielded good results in previous weeks.

Unbalanced Dataset

The uneven label distribution is certainly affecting the overall performance of the classifiers and will be resolved either by incorporating another, more evenly-balanced, dataset provided by the CLEF Team, or by over-sampling and, respectively, under-sampling the data at hand.

BERT Team

Unbalanced Dataset

Similarly to the 3Layer Team, this team decided to eliminate some of the *False* labeled news in order to make the label distribution more even.

4-Way Algorithm

It was later discovered that the **num_labels** parameter solves the problem, thus, during the next week's run, the implementation will be enriched this way.

Insufficient RAM

To solve this issue, the team will have to find the maximum value of the **max-length** parameter when encoding the tokens, as well as the right batch size and number of epochs.