Stance Detection for the Fake News Challenge Dataset using Deep Learning

Anak Agung Ngurah Bagus Trihatmaja and Shishir Kurhade

Abstract

The problem of fake news has arisen recently as a threat to high-quality journalism and well-informed public disclosure. The goal of fake news challenge is to explore how artificial intelligence technologies, particularly machine learning and natural language processing, might be leveraged to combat the fake news problem. The goal of our project is to develop machine learning models to predict a stance label (Agrees, Disagrees, Related, Unrelated) with respect to the title for a respective news article. For this purpose we will use gated recurrent unit (GRU) to predict the labels. As a baseline to measure performance we will also solve the problem using logistic regression method.

1 Introduction

In this project, we try to combat a serious problem in our media using machine learning techniques. Fake news has been accused of increasing political polarization and partisan conflict in the United States during the divisive 2016 presidential campaign and also during the Brexit referendum in 2016. This problem views the task of fake-news detection as a stance detection problem which is a labelling task. We want to automatically classify a news into four labels, which are unrelated, agrees, disagrees, and discusses.

A reasoning for these labels is as follows:

1. **Agrees**: The body text agrees with the headline.
2. **Disagrees**: The body text disagrees with the headline.
3. **Discusses**: The body text discuss the same topic as the headline, but does not take a position
4. **Unrelated**: The body text discusses a different topic than the headline

The classifier that we build could later be used as a base of a fake news detection tool that can automatically categorize the news into the stances given.

2 Proposed Project

2.1 Dataset Overview

The data provided by the Fake News Challenge consists of headline, body, and stace. For training, there are two csv files:

1. *train_bodies.csv*: contains the body text of articles with its ID
2. *train_stances.csv*: contains labeled stances for pairs of article headlines and article bodies, in which the article bodies rever to the bodies in train_bodies.csv

The distribution of the data is as follows:

Rows	Unrelated	Discuss	Agree	Disagree
49972	0.73131	0.17828	0.0736012	0.0168094

Roughly, we will use 2000 samples as our development set, for choosing hyperparameter and performance evaluation, and leave the rest for our training data.

2.2 Methods

2.2.1 Pre-processing

The text from the corpus will be converted to tokens using *nltk* package and then be mapped to corresponding vectorized forms using pre-trained *GloVe* representations freely available on the Stanford NLP group website. Since the text sequences observed will be of variable length we will pad all sequences to the length of the maximum length text sequence before inputting it to our model. The data pre-processing includes normalising the case, handling the punctuation and non-alphabetic symbols.

2.2.2 Learning Model

We will used a learning model that is RNN variant to predict the stances. The reason why it is based on RNN instead of CNN is that, for language modelling, RNN models are still the best approach (Mikolov et al. 2011). The difficulty with RNN models is that they are hard to train, because they suffer from the vanishing gradient problem (Kevin P. Murphy book page 570). The solution to this problem is to use RNN variant called long short-term memory (LSTM). However, recently a model called gate recurrent unit (GRU) was introduced by Cho et al. [2014]. GRU is similar with LSTM. Unlike LSTM, GRU combines forget and input gates into a single update gate and merges the cell state and hidden state. This makes GRU computationally more efficient than LSTM and GRU model has been increasingly popular. For this project we will use TensorFlow API for GRU, tf.nn.rnn_cell.GRUCell.

It is often said, that neural network performs better for a case like this. Therefore, in this chance, we will also perform logistic regression as a baseline for our project. In this project, we will build the logistic regression from scratch.

To measure the accuracy, we will use the evaluation tool provided by Fake News Challenge. The tool will evaluate our model and output a score. The score is weighted as follows:

1. Classifying pair of body and headline as related and unrelated is weighted 25%
2. Classifying related pairs as agrees, disagrees, or discusses is weigted as 75%

The related/unrelated classification task is expected to be much easier while classifying the agrees, disagrees or discuss is more difficult and more relevant to fake news detection.

3 References