Fake News Detection Tool: How to build a machine learning model that distinguishes between real news and fake news for users to find helpful?

Semester 1 Report

Kārlis Siders

2467273S

Proposal

Motivation

The purpose of the project is to build a usable machine learning (ML) model that distinguishes between real news and fake news since 1) the spread of fake news online is spreading quickly and increasing in amount (especially since Elon Musk's takeover of Twitter), and 2) people in general are very bad at telling these two types of "news" apart[1]. This poses a dangerous problem of spreading misinformation and distrust of the more reputable sources found online, which increases polarisation and radicalisation.

Aims

- Build a machine learning model that
    - uses multiple features of a given Tweet,
    - applies the latest ML developments in all parts of the processing, e.g., BERT for pre-processing, and
    - has 70%+ F1 score for distinguishing between real and fake news.
- Build a website that
    - connects the model built previously with the front end for a user to interact with,
    - (optionally) displays confidence for estimated class for a given input,
    - applies the latest front-end developments, e.g., Bootstrap for best-looking User Interface, and

[1] As I will mention (and cite) in the dissertation, a study says that 54% of times online users could not tell the two apart.

    - is hosted on a server/domain accessible by everyone, i.e., not just university students.

Progress

So far, I have:

- developed simple models for binary classification with a limited dataset (330 Tweets) of Twitter fake and real news,
- found a bigger dataset (420,000 Tweets) to use for analysis and imported its data straight from Twitter API, the access to which had to be requested and which had to be learnt,
- fixed multiple problems with the dataset, including grouping of Tweets by their article (which, before, caused the models to perform unrealistically well), cleaning of unnecessary information, and fixing the lack of balance between classes,
- experimented and learnt much about tokenisation and BERT pre-processing, and
- built simple models for analysis of Tweets just by their source text with 66% weighted F1 score.

Problems and risks

Problems

- Time management since it had been difficult to focus on this long-term project when assessed exercises were more pressing for multiple weeks;
- Difficulty of learning about machine learning, a specific API, and front end while trying to achieve the tasks set in the project;
- Fixing problems in the dataset outlined in Progress.

Risks

- Problem: Dividing the time between dissertation, other university courses, work, and social life.
    - Mitigation: Choosing a specific day to only concentrate on the project (both dissertation and code) and working on the project for an hour every other day.
- Problem: The model will be overfitted to the training data and not be very useful to the end user.
    - Mitigation: Using cross-validation and regularly checking the models with “fake news” and real news sentences as input.

- Problem: Twitter API might stop working because of changing times for Twitter.
    - Mitigation: Provide a second, simpler, machine learning model on the website as a back-up, which only needs text input that can be copied into a submission text area.

Plan

- December:
    - Week of 12th – 18th: Research reason for currently skewed matrices, start applying BERT pre-processing of larger dataset by batches
    - Week of 19th – 25th: Finish BERT pre-processing, add multiple features in pipeline
- Week of 26th Dec – 1st Jan: Build basic Django website that is connected to the best-performing model with multiple features and add the option of using the simpler (text-only) best-performing model.
- January:
    - Week of 2nd – 8th: Use Twitter API for model input being the URL of a Tweet (for the more feature-rich model), add Bootstrap for prettiness, and host website on www.pythonanywhere.com.
    - 2 weeks (9th – 22nd): Write Introduction.
- 2 weeks (23rd Jan – 5th Feb): Write Background.
- February:
    - Week of 6th – 12th: Write Analysis/Requirements.
    - 2 weeks (13th – 26th): Write Design.
- 2 weeks (27th Feb – 12th Mar): Write Implementation.
- March:
    - Week of 13th – 19th: Write Evaluation.
    - Week of 20th – 24th: Write Conclusion.