A fake news detector

Proposal for the Capstone Project
Udacity Machine Learning Engineer Nanodegree

Submitted: November 2, 2017

Author: Stefano Casasso

1 Domain background

As opposed to *real news*, which is based on true facts and has the goal of informing the audience, *fake news* can be defined as "a made-up story with an intention to deceive" [1]. Fake news is probably as old as real news, but it has become a major problem with the coming of the web which made it extremely easy and fast to circulate contents among a potentially very large number of people. In particular, with the success of social networks in the last decade, fake news are spread around like a virus by people, usually with limited education and background, who share them in their public profiles.

2 Problem statement

The problem under study in this project is to discriminate between real news and fake news.
The underlying idea is that this differentation is possible by systematically analyzing the content of the news, both in terms of text and headline. The goal is to develop an algorithm that takes as input the content of the news and gives as output a label which can be "real" or "fake". This is a *binary classification problem* which will be studied using supervised machine learning techniques.

3 Datasets and inputs

In order to successfully apply supervised learning techniques, we need a sizeable "labelled" dataset, preferably with a good balance between the different classes in output (fake, real).
In the following the strategy to gather these data is outlined. Since the data for the two classes come from different sources, they will be merged together using a common subset of the columns. However, as already mentioned previously, there is no plan (at this stage) to use any other information than the body and the headline.

Fake news dataset

This part, which is the most challenging, is fortunately already publicly available. The popular data science platform Kaggle provides a dataset, in form of .csv file, which contains about 13000 fake news [2].
This dataset is built by crawling the news from 244 websites which are tagged as not reliable sources by the BS Detector project. BS Detector ultimately uses a manually compiled list taken from the OpenSources platform. As does Kaggle, we consider this an authority in the domain of fake news and we assume that these data are purely fake news.

The dataset consists of 20 columns, among which there are of course `title`, `text`. The use of additional variables is not planned and depends also on the availability of the same type of information also for the real news dataset. All the news in the dataset are collected in November 2016.

Real news dataset

For the "real" news, we use the freely downloadable dataset from webhose.io, under "English news articles". All the news come from November 2016, thus matching perfectly the time frame of the fake news data described above. The dataset is huge and consists of about 500k news

articles, in the (quite inconvenient) format of a single json file per news. We apply an additional processing on top of these corpus, which consists of producing a single file and restricting the sources of news to the followings: *New York Times*, *Washington Post*, *The Washington Journal*, *Reuters*, *The Guardian*, *Forbes*, *BBC*, *NPR* and *Bloomberg*. These websites are universally recognised as trustable sources of information and we are confident that they represent an unbiased sample of real news.
As the number of news selected this way it's still much larger compared to the fake news dataset ($\sim$100k vs. $\sim$13k), we further skim the corpus by "capping" each source to a maximum of 3k articles, chosen randomly. This procedure ensures that there is no source which dominates the others, thus making the dataset more balanced (we noticed that a sizeable fraction of news come from Reuters alone). After this additional step we are left with about $\sim$24k articles.

4 Solution statement

The solution to the problem described in 2 consists of building a model which successfully identifies and labels fake news taking as input the news headline and body text.
More details about the steps that have to be implemented to achieve this are given in 7.

5 Benchmark model

As mentioned before, some implementations of a fake-news detector are already present in the web, see for instance [3, 4]. In these example, accuracy of at least 80% are achieved on the test dataset. We aim at reproducing similar results. However, even though the fake news dataset is the same, the real news dataset is collected in a different way: for this reason the performance are not, strictly speaking, completely comparable.

6 Evaluation metrics

The size of the data is different among the two classes, as the real news are almost the double of the fake news. For this kind of situation, the best metric is the F1 score which is defined as the harmonic average of the *precision* and the *recall*. In this study the precision is defined as the number of fake news labelled as fake divided by the total number of news labelled as fake. The recall is defined as the number of fake news labelled as fake divided by the total number of fake news in the dataset.
The F1 score will be used also for the optimization of the model hyperparameters of the model.

7 Project design

A tentaive ordered workflow is given below.

1. **Preparing the dataset**. As the data from the two output classes come from different sources, they have to be merged in the same structure.

2. **Converting text to features**. This part involves the application of some *Natural Language Processing* (NLP) techniques to convert the body text to a vector of numerical features which can be fed to a supervised learning model. The baseline approach that is

considered here is the so-called *TF-IDF* [5], which assigns a weight to each word in the document, for each document in the corpus, according to the importance of the word in the document.

3. **Choosing the model(s)**. Once each article has been "vectorized" in a space of features, a machine learning algorithm for classification can be chosen and trained on the data. We plan to test more than one model and compare them using the specified metrics (more details below). Among the algorithm which have been successfully used [3, 4] there are: *naive bayes*, *logistic regression*, *random forest* and and *support vector machine*. We plan to test at least 3 of them.

4. **Choosing appropriate performance metric**. More details are given in 6.

5. **Fine-tuning the hyper-parameters**. Having chosen the model and the performance metrics, the search for the best model will be carried out using the usual grid search across the hyper-parameter space.

6. **Summarize the results**. As a final step the results are summarized, corresponding to the different techniques used and the corresponding accuracy achieved.

The step as of 1 will be done within a script which will be also included for reproducibility purpose. We plan to include all the other steps in the same *Jupyter Notebook*.

References

[1] S. Tavernise, "As fake news spreads lies, more readers shrug at the truth." https://www.nytimes.com/2016/12/06/us/fake-news-partisan-republican-democrat.html, 2016.

[2] Kaggle, "Getting real about fake news: Text and metadata from fake and biased news sources around the web." https://www.kaggle.com/mrisdal/fake-news, 2017.

[3] Y. Genes, "Detecting fake news with nlp." https://medium.com/@Genyunus/detecting-fake-news-with-nlp-c893ec31dee8, 2017.

[4] J. Goldstein, "Identifying "fake news" with nlp." https://blog.nycdatascience.com/student-works/identifying-fake-news-nlp/, 2017.

[5] Wikipedia, "tf-idf." https://en.wikipedia.org/wiki/Tf-idf, 2017.

[6] A. H. Bloomberg, "Facebook has a new plan to curb 'fake news'." http://fortune.com/2017/08/03/facebook-fake-news-algorithm/, 2017.