"Make the truth great again" - Articles' Reliability Determiner

Team Anh-Thi - Term Project Proposal for CS574 - Spring 2018

Ngo Sy Toan, Nguyen Tuan Anh, Vu Phuong Thao

1 INTRODUCTION

1.1 Our Team

Team Anh-Thi consists of 3 undergraduate students, of which two are from KAIST (Korea Advanced Institute of Science and Technology) and one is from UTC (University of Technology of Compiegne, France). This proposal is for the Term Project of NLP class in 2018 Spring, taught by Prof. Key Sun-Choi. Role of each member is provided as follows.

- **Ngo Sy Toan:** Feature generating, classifier testing.
- **Nguyen Tuan Anh:** Feature generating, classifier implementing.
- **Vu Phuong Thao:** Text preprocessing, classifier implementing.

1.2 The Problem

According to Facebook's record, the number of engagements for top 20 election stories in US 2016, fake news exceeded mainstream news for 1.4 million. Defined as "a made-up story with an intention to deceive", fake news has become increasingly popular and caused a great deal of confusion about current events. Widely spreading misinformation on all social media, fake news potentially manipulates public perception and mislead a whole community.

1.3 The Dataset

The dataset is published on Kaggle, contains features of ID, title, author, and text. For the training set, a binary feature of reliability is also given. The training set contains 20800 instances while the test set has 5200 instances. Our purpose is to determine whether a given news is reliable or not based on above features.

2 RELATED WORK

The main reason making detecting fake news technically challenging is even human may not be able to distinguish between a fake and a real post. There have been many efforts to help audiences confirm the reliability of a post such as fact-checking websites, but they can hardly return an immediate response. Besides, the number of news we encounter every day from sources are countless, which is troublesome to manually look up for each event in detail.

Therefore, research has been conducted to detect fake news automatically based on linguistics and machine learning approach. Each research group has their own definition of "fake news", ranging from click-bait to rumor. In our research, we consider the reliability of the post given its content, author, title.

The main target of our project is to work on text data, majorly the content of the news. It has been proved that there are certain connections between linguistics structure and components, such as conjunctions, pronouns, and the reliability of the post [1] [2]. However, fake news of different topics may be indicated by different groups of terms and structures, so to efficiently detect the relevant patterns, one may need to work deeper on semantic analysis for useful linguistic indicators. Due to the large bound of the content of the dataset, we decide not to work much on the semantic pattern.

As fake news detection deals with a large amount of text and high-dimension vectors, Deep Learning has been used by many researchers and proved to work well on text data [3], namely Recurrent Neural Networks(RNN). With the ability to process a sequence of input by using internal state, RNN especially works well on time series [4] and text data.

3 APPROACH

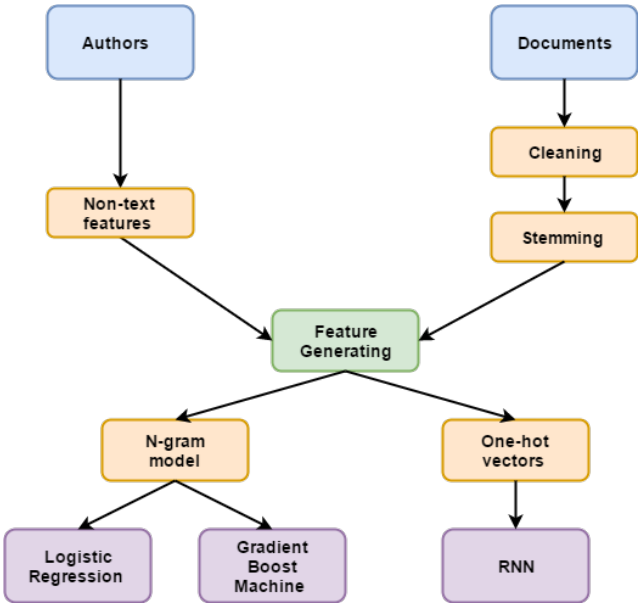


Figure 1: Approach

3.1 Preprocessing

For preprocessing, we focus on the given text data, namely the news content and the title. In detail, these processes will be applied:

- Tokenizing: Documents are turned to lower case then converted into word-vectors.
- Punctuation cleaning: Removing defined punctuation from word-vectors.
- Stopword cleaning: Removing words which contain no information such as "the","to".

Natural Language Processing, Spring 2018, KAIST

- Stemming and lemmatizing: Transforming words toward their root form.

All of these functions are provided in the nltk package of Python.

3.2 Model and text feature generating

We consider not only single words in documents but also important phrases. Therefore, we propose an n-gram feature approach. Unigram, bigram, and trigram are extracted from the documents and feed to the model. Several different models will be implemented for comparing and each model has its own characteristic and fits well with the different type of features. Understanding this issue, for each model, we design a different feature extraction method as follow:

Logistic Regression(LR): LR is a statistical method which analyzes a dataset and gives the output which only has two possible value. The reason we choose LR is how it returns output and its simplicity. In case of LR, n-grams will be fed into the model with TF-IDF vectorizing. Besides, a new promising approach may be applied, namely, a filter layer is added to the model before this layer forwards data to the sigmoid node. This filter helps to reduce the dimension of feature and extract important and relevant one. The activation function of this layer is still under consideration.

Gradient Boost Machine(GBM): GBM is a machine learning technique for regression and classification problem, which ensembles the prediction of weak models to a stronger and more accurate model. We use package **XGBoost** for GBM implementation. XGBoost is an open source software library which provides the gradient boosting framework and has gained popularity and attention recently as it was the algorithm of choice for many Kaggle competitions. In case of GBM, n-gram will be fed into the model with bag-of-words method vectorized by TF-IDF.

Recurrent Neural Networks(RNN): RNN is a class of artificial neural networks which has been proved to be able to capture contextual information. In RNN, inputs are fed to model in sequence, so a node can know some information about previous nodes. Therefore, RNN already has the characteristic of n-gram model in somewhat level. N-gram only shows their effectiveness on RNN when n is large enough. Instead of bag-of-words with n-gram model, we will feed the model with a one-hot vector at each node.

3.3 Non-text feature generation

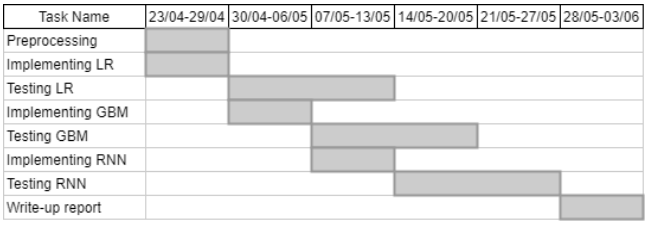
Even though the text feature plays the key role in reliability determining, we also make use of the non-text feature namely author. Naturally, authors who have unreliable posts in training set are more likely to spread more misinformation. Besides, the set of authors in test set slightly differs from the set of authors in training set. Therefore, we can assign each author with a score based on the number of unreliable and reliable post they wrote. Namely, authors with decent posts will be assigned a positive score and ones writing fake news will get a negative score. A relatively small number of authors who aren't mentioned in the training set will be initialized with score 0.

The RNN model requires one-hot encoded features, so we add 3 more bits to the dictionary which represent for the reliability of the authors (reliable, unreliable and unknown). The non-text feature will be added as a node before text feature in training, and in that way, it can contribute to the whole text behind.

4 EVALUATION

Evaluation metric: as we have a binary output (the reliability of an article), we use F1 metric for evaluation. Previous works testing on different datasets achieved an accuracy of 91% [1] and 90.1% [3]. We did a quick experiment with TF-IDF and Random Forest, which gave an accuracy of 89%. Therefore, our goal is to build a model pushing the accuracy to 92% - 95%.

5 TIMELINE



REFERENCES

[1] Vanessa Wei Feng and Graeme Hirst. 2013. Detecting Deceptive Opinions with Profile Compatibility
[2] David M Markowitz and Jerey T Hancock. 2014. Linguistic traces of a scientic fraud: e case of Diederik Stapel. PloS one 9, 8 (2014), e105937
[3] Natali Ruchansky, Sungyong Seo, Yan Liu. CSI: A Hybrid Deep Model for Fake News Detection CIKM '17 Singapore
[4] Michael Husken and Peter Stagge. 2003. Recurrent neural networks for time ÂÍ series classication. Neurocomputing 50 (2003), 223-235