COMP9417 Final Project

Topic 1.5: Learning to predict ”stance” of news articles

Thanet Sirichanyaphong
z5228028

Suphachabhar Nigrodhananda
z5176892

11 August 2019

1 Introduction

Nowadays, people use social media as a function to communicate with others including spreading and receiving news. It’s easy to post or share somethings that we are interested without considering the information correctly. Some of them are not a credible source and this can lead to a big problem in our societies. This prompted academia and industry around the world to create the Fake News Challenge(FNC) to explore how artificial intelligence technologies, particularly machine learning and natural language processing, might be leveraged to combat the fake news problem [1]. Obviously, this is a difficult and complex problem to solve. Fortunately, the challenge broke down the process into several steps. The purpose of this study is to design an intelligent stance detection of a body text from a news article has agree, disagree, discuss or be unrelated relation to a headline. Then, we evaluate its performance and compare the result with the FNC stage-1 competition.

2 Classification architecture

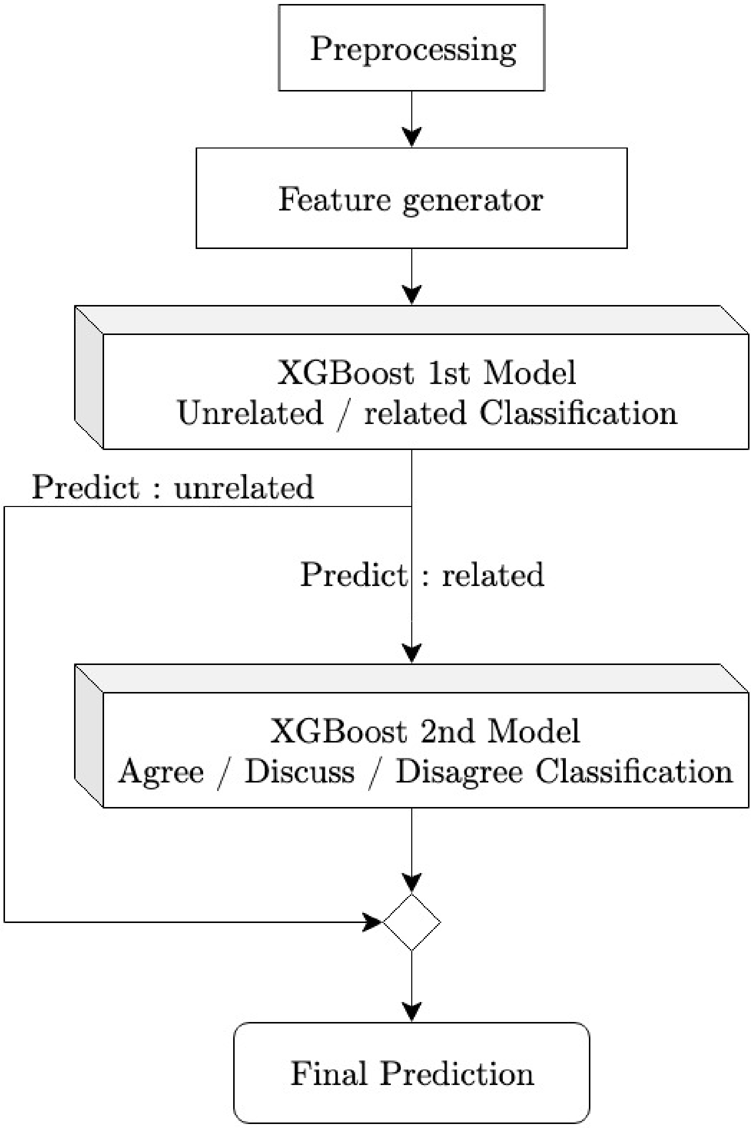

Figure 1: classification architecture

3 Implementation

Our implementation is available from `https://github.com/aiThanet/FNC-1`.

Approach

1. Firstly, we trained 2 XGBoost models. The first one used for unrelated/related classification. The second one used for agree/discuss/disagree classification.
2. We didn't tune the hyperparameters for both models due to the limitation of resources and time. The majority of time spent on feature engineering because some features need several hours to generate the result.
3. In addition to FNC-1 baseline features [3], we also implemented four more features: sentiment features, name entity features, question mark ending feature as suggested in [2] and paragraph vector as suggested in [4].
4. Although the testing set label has already released, the models have been trained and tuned only from the training set.
5. To predict an input, it must be classified by the first model. If it is predicted to be related. Then, the second model will identify the relation between its headline and body. The structure of the system is shown in Section (3).

Proprocessing

Data pre-processing technique we used in our project are punctuation removal, converting to lower case, stop words removal and lemmatization. Each feature has a different data pre-processing style, more detail of each feature will be described below.

Feature description

1. FNC-1 Baseline features:
   - `overlap` is $\frac{number\_of\_word\_in\_headline\_intersect\_body}{number\_of\_word\_in\_headline\_union\_body}$. Each word has been removed punctuation, lemmatized and converted to lowercase.

- `refuting` is a list of each refuting word is in a headline or not.
- `polarity` is the polarity of headline and the polarity of body.
- `hand` is consist of number of unigrams in body, number of unigrams without stop word in the body, number of {2, 4, 8, 16}-chargrams of headline in body and number of {2, 3, 4, 5, 6}-ngrams of headline in body.

2. Sentiment features:
   These features use the Sentiment Analyzer in the NTLK package to separately assign the polarity score to the headline and body. There are 4-types of score: compound, negative, neutral and positive which represents the polarity of the sentence.

3. Name entity similarity features:
   These features use the Named Entity Recognition package from Stanford CoreNLP to find which text has Person, Location, Organization entity. Then, we compute the similarity and distance of headline and body. The similarity of each entity is $\frac{number\_of\_entity\_in\_headline\_intersect\_body}{number\_of\_entity\_in\_headline\_union\_body}$. The distance of each entity is $1 - similarity$.

4. Question mark ending feature:
   Whether the headline ends with a question mark.

5. Paragraph vector features:
   Although extracting features from the headline and body are widely used Bag of Words model, distributed word vector techniques published in [4] have been shown to outperform Bag of Words model. In this paper, an algorithm called Paragraph Vector show the better result. These features are generated from 2 doc2vec models in the gensim package. Each model used to assign the paragraph vector for headline and body separately.

Model description: XGBoost
We train 2 XGBoost models, the first one is trained by all stances but for the second one, training data only come from the related stances. Both models use default parameters.

4 Result

We followed the scoring system in the Fake News Challenge [1] to evaluate the model. The final result is shown in Table 1 and Table 2, the score of testing data set is 9069.25. According to the competition leader board on `https://competitions.codalab.org/competitions/16843#results`, our rank is 20th. Furthermore, confusion matrix of development and testing data set are reported in the table.

Scores on the development set				
	agree	disagree	discuss	unrelated
agree	244	0	452	66
disagree	29	3	119	11
discuss	89	7	1543	161
unrelated	9	0	117	6772
Score: 3657.0 out of 4448.5 (82.207%)				

Table 1: Confusion matrix on development set

Scores on the testing set				
	agree	disagree	discuss	unrelated
agree	701	6	1009	187
disagree	119	11	383	184
discuss	551	16	3450	447
unrelated	102	6	696	17545
Score: 9069.25 out of 11651.25 (77.839%)				

Table 2: Confusion matrix on testing set

We also compared the methods we have done for the experiment and our final classifier as shown in Table 3. We performed feature increment testing on an XGBoost model that classify 4 classes and our final classifier. Then, comparing the score from each method.

method	feature used	dev_set score	testing_set score
XGBoost	baseline(1)	79.167%	75.016%
XGBoost	1,2	79.830%	75.241%
XGBoost	1,2,3	80.179%	75.460%
XGBoost	1,2,3,4	80.094%	75.430%
XGBoost	all	80.724%	76.945%
final	1,2	80.482%	76.736%
final	1,2,3	81.505%	77.017%
final	1,2,3,4	81.572%	77.249%
final	all	82.207%	77.839%

Table 3: Comparison of method

References

[1] Fake news challenge stage 1 (fnc-i): Stance detection. `http://www.fakenewschallenge.org`, 2016.

[2] William Ferreira and Andreas Vlachos. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1168, San Diego, California, June 2016. Association for Computational Linguistics.

[3] Byron Galbraith, Humza Iqbal, HJ van Veen, Delip Rao, James Thorne, and Yuxi Pan. Baseline fnc implementation. `https://github.com/FakeNewsChallenge/fnc-1-baseline`, 2016.

[4] Quoc V. Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. *arXiv e-prints*, page arXiv:1405.4053, May 2014.