TECHNICAL REPORT

Fake news detection:

An Approach based on Detecting the Stance of Headlines to Articles

Problem Statement & Objective : The exponential increase in fake news generation and distribution of inaccurate news creates an immediate need for automatically tagging and detecting such manipulated news articles. However, it is difficult to automate the fake news detection without human intervention as it requires the model to understand nuances in natural language. Moreover, the majority of the existing fake news detection models treat the problem at hand as a binary classification task, which limits the model's ability to understand how related or unrelated the reported news is when compared to the real news. To address these gaps, we have designed a framework where we integrate the machine learning model and transformer model to accurately predict the stance between a given pair of headlines and article bodies.

I. Experiment-1

AIM: To understand how the Machine learning model classifies headlines and articles into one of the 4 target variables (agree, disagree,discuss and unrelated) .

METHOD : Text preprocessing, stop words removal and features such as GloVe Similarity, K-L Divergence, N-Grams(code) were used as predictors in the ML model for 4-class prediction. We used the FNC-1 dataset (75385 entries) on 11 different machine learning models- SVM classifier, SGD classifier, Random Forest classifier, Logistic regression, Decision tree classifier,KNN,Quadratic discriminant analysis, Linear discriminant analysis, Gaussian Naive bayes,Adaboost classifier, and Xgboost classifier.

CONCLUSION : Logistic Regression, SVC classifier and SGD classifier gave nearly 87% model accuracy compared to other models but observed high misclassification rate about 14% .During this experimental stage model fails to well differentiate the articles into one of the 4 target variables.

CODE : Python code

II. Experiment-2

AIM: To construct a Machine learning model for 3-class prediction(agree, disagree and discuss) and to make the model learn to well differentiate the articles between related labels.

METHOD : To make the model learn more about the relation between headline and articles , we used NLP techniques to extract more features- Jaccard similarity, Jaccard similarity noun, n-grams, LDA score (FeatureExtractionCode),KL divergence, semantic similarity,GloVe similarity. These 7 features in different combinations as predictors were incorporated into different ML models for 3-class predictions.

CONCLUSION : The best combination of features were, when all the 7 features as predictors gave best results of 71% accuracy using Random forest classifier compared to all other ML models. Even the Experiment-2 fails because the model had a 29% misclassification rate during prediction.

CODE : Python code

III. Experiment-3

AIM: To construct a Machine learning model for 2-class prediction(related and unrelated) and to reduce the misclassification rate.

METHOD : We use all the 7 features - Jaccard similarity, Jaccard similarity noun, n-grams, LDA score,KL divergence, semantic similarity,GloVe similarity in different combinations as predictors and were incorporated into different ML models such as SVM classifier, SGD classifier, Random Forest classifier, Logistic regression, Decision tree classifier,KNN,Quadratic discriminant analysis, Linear discriminant analysis, Gaussian Naive bayes,Adaboost classifier, and Xgboost classifier for 2-class predictions.

CONCLUSION : The best combination of features turned out to be Jaccard nouns similarity, GloVe similarity and N-grams. These 3 features gave 81% accuracy using Random forest classifier compared to all other ML models and even the misclassification rate was reduced to 18%. During this experimental stage , we noticed the improvements in classifying the articles into related and unrelated labels .Hence , as a result we proposed our Baseline model-1 to be RandomForest classifier for 2-class prediction.

CODE : Python code

IV. Experiment-4

AIM: To construct a model that captures the contextual meaning of the sentences in the article. The BERT classification model for 4-class prediction(agree, disagree,discuss and unrelated) was used to understand the performance of this model before going into 3-class prediction.

METHOD : We use pytorch framework for training the BERT classification model. BERT uses a 'base-uncased' pre-trained model that were on 110M parameters. Using this pre-trained model, we downstreamed the model to classification task by fine-tuning the model parameters:

1. model class: "Classification Model"
2. model type: 'Bert'
3. model name: ' bert-base-uncased'
4. num labels: 4
5. args:{'learning_rate':3e-5, 'num_train_epochs': 5, 'reprocess_input_data': True, 'process count': 10, 'train_batch_size': 4, 'eval_batch_size': 4, 'max_seq_length': 512, 'fp16': True}

After fine-tuning, the same FNC dataset which consists of 64205 entries was used. From this only Headlines-articles and target variables without stop word removal were given as an input to the model.

CONCLUSION : BERT model on 4-class labels gave 96% on validation set and 91% on test data. The BERT model performed very well on 4-class prediction with less training loss of 0.18. Hence, we can implement this model on 3-class predictions.

CODE : Python code

V. Experiment-5

AIM: To perform a 3-class prediction using the BERT classification model.

METHOD :Again fine-tuning the num_labels parameter to 3-class:

1. model class: "Classification Model"
2. model type: 'Bert'
3. model name: ' bert-base-uncased'
4. num labels: 3
5. args:{'learning_rate':3e-5, 'num_train_epochs': 5, 'reprocess_input_data': True, 'process count': 10, 'train_batch_size': 4, 'eval_batch_size': 4, 'max_seq_length': 512, 'fp16': True}

After fine-tuning, the same FNC dataset which consists of 23000 related articles was used to train the model. From this only Headlines-articles and target variables without stop word removal were given as an input to the model.

CONCLUSION : BERT model on 3-class labels gave 99% on training set ,88% on validation set and 75% on test data. The BERT model performed very well on 3-class prediction with less training loss of 0.06 (ModelOutput) , Matthew's correlation coefficient 0.97 and overall misclassification rate was only 1,6%.The BERT model performed the task with minimum loss and turned out to be a good model.Hence, we proposed this model as Baseline model-2 for 3-class predictions.

CODE : Python code

VI. Experiment-6

AIM: To integrate Baseline model-1 with Baseline model-2 as our complete solution framework for Fake news detection and test this model on real-time data.

METHOD : 175 claims of recent time were collected and we built a web scraper for scraping the articles for those claims. A Total of 2000 articles were extracted . In order to avoid information leakage in the Bert model(limited to 512 input tokens), we used the BART summarizer model to reduce the article length by capturing the entire contextual meaning of the article(code) .BART (Bidirectional Auto-Regressive Transformer) is limited to 1024 input tokens Then claims as headlines and summarized articles were given as input to Baseline model-1 was able to predict more related articles for 126 claims out of it 46 claims were fake news and 80 claims were real news.Out of 49 claims which predicted more unrelated articles ,33 claims were fake news and 6 claims were real news. Next, for Baseline model-2 we only considered related articles for 3-class prediction. Later, we came up with the concept of threshold to output the final predicted results into Fake or Real news. Here, we used two conditions: (1) If the claim has predicted frequency count less for agree labels than compared to the sum of disagree and discuss labels and (2) If frequency count of discuss is less than disagree, then in both the cases we output that claim as Fake news, or else Real News.

CONCLUSION : As a result of imposing conditional criteria, 55 claims out of 74 fake claims were correctly classified as fake news and on other hand only 26 claims out of 101 real claims were classified as real news. The reason for the high misclassification rate in real news is because we tried to make our model output biased towards predicting fake news correctly. We can do the same for real news as well but since our objective was only to identify the fake news correctly in the initial phase.Further, it can be continued as a part of our future work Hence, At this stage we restrict our work to 3-class prediction.

CODE : Complete Pipeline for FakeNewsDetection