CS229 Spring 2021 Final Project

Comparing Machine Learning Techniques on the Task of Fake News Detection

Name: Qianli Song (qslong), Yining Zhu (zhuy0713)

1. Introduction and Related Work

In our project, we explore the effectiveness of different machine learning methods in detecting fake news. The input is frequency of words in true or fake news. The output is 1 or 0, where 1 represents true news and 0 represents fake news. The model that we use to predict the output includes: Naive Bayes, Support Vector Machine, Convolutional Neural Network and Bert. There're five papers for the related work. 1) Detecting Fake News with Machine Learning Methods. [1] Like our project, this paper uses Naive Bayes, Neural Network and Support Vector Machine. The difference is that the data of this paper comes from Twitter message, and its model uses multiple features such as FriendsCount and FollowersCount, rather than just the words of the message. Its strength is that it's concise and clear, and achieves high accuracy rate. The weakness is its data is only from Twitter. 2) Detecting Fake News using Machine Learning:A Systematic Literature Review.[2] Its strength is that this paper discuss a good number of machine learning classifiers from multiple research literatures in detecting fake news. 3) Supervised Learning for Fake News Detection.[3] Its strength is that it explores a great number of feature types, and presents a new set of features for automatic detection of fake news. The weakness is that its data is only from Buzzfeed News. Its difference from ours is that this paper not only explores language features (bag-of-words), but also uses lexical features, psycholinguistic features and semantic features in detecting fake news. Besides Naive Bayes and Support Vector Machine, this paper also uses XGBoost and k-Nearest Neighbor. 4) Which Machine Learning Paradigm for Fake News Detection.[4] The strength is that this paper presents a comprehensive performance evaluation of eight different machine learning algorithms and also explores three separate datasets. It also features in-depth performance evaluation and analysis. 5) A Benchmark Study on Machine Learning Methods for Fake News Detection. [5] Its strength is that it uses diverse data sources, which also includes LIAR. It also uses several feature extraction approaches, including Lexical and Sentiment, n-gram, Empath Tool and Pre-trained Word Embedding.Similar to our project, this paper also uses several different machine learning approaches. This paper has in-depth performance evaluation in the end.

3. Dataset and Features

Datasets

We have used three datasets.**The first dataset** (later referred to as Kaggle I)[6] is the Fake and Real News dataset from Kaggle (see link below). This dataset contains one true news file and one fake new file. The true news file contains 21418 pieces of news scrapped from Reuters, a source of true news. The fake news file contains 23482 pieces of news scrapped from 21 Century Wire, a source of fake news. The news are from 2016-2017.**The second dataset** (later referred to as Kaggle II)[7] consists of news articles from various websites. It has 2006 different pieces of news. It contains 1740 pieces of data **The third dataset** (later referred to as Liar)is LIAR, A Benchmark Dataset For Fake News Detection. [8] This dataset has training, validation and test dataset. This dataset contains 5490 pieces of data.

-**Training and validation data:** Kaggle I is our main data for training and validation, since the size of the data is significantly larger than the other two. For more diversity in the data, we also added Kaggle II and Liar data into the training set. Because the size of these two dataset is really small, we did not train models based on only these two datasets. For each type of algorithm, we trained two models, one with only Kaggle I data, and the other with all the data combined. For each dataset, 80% of the data are used for training, and 10% of the data are used for validation.

-**Test data:** 10% of each dataset are used for testing the models.

Features

Bag-of-words: we converted the list of news articles into a matrix. Each column represents a unique word in the vocabulary or all articles, while each row represents an article. The value of each element indicates the number of times a word occurs in an article. Since rare words are not useful for modeling, we only add words to the dictionary if they occur in at least fifteen pieces of news.
Tf-idf: we also used td-idf features, which would normalize the data on the different lengths of the news articles, as well as the frequency of the word across all articles.

4. Methods

(a) **Naive Bayes** We use Multinomial Event Model and apply Laplace Smoothing.The maximum likelihood estimates of parameters: $\phi_{k|y=r} = \frac{1+\sum_{i=1}^{n}\sum_{j=1}^{d_i} 1\{x_j^i=k \wedge y^{(i)}=r\}}{|V|+\sum_{i=1}^{n} 1\{y^{(i)}=r\}d_i}$, r is 0 or 1. $\phi_y = \frac{\sum_{i=1}^{n} 1\{y^{(i)}=1\}}{n}$.

(b) **Support Vector Machine** We implement the SVM algorithm using the LinearSVC function in scikit-learn. The algorithm finds a hyperplane that best divides the data points according to their labels. The cost function for the margin between the data points and the hyperplane is

$$c(x, y, f(x)) = \begin{cases} 0 & \text{if y * f(x)} \geq 1 \\ \text{1-y*f(x)} & \text{otherwise} \end{cases} \tag{1}$$

(c) **Neural Network** We use an open-source deep-learning framework called Keras, from TensorFlow. In Keras, we use the sequential model, which means that it uses plain stack layers and each layer has exactly one input tensor and one output tensor. There are four layers in total. The first three layers are Conv1D layers (1D convolution layer) with input shape 128, 256 and 512 respectively and kernel size 4. These three layers use the ReLU (Rectified Linear Unit) activation function. We have also applied batch normalization to these three layers. The fourth layer is the Dense layer(densely-connected Neural Network layer) and it uses the sigmoid activation function.

(d) **BERT** BERT is a general-purpose language-understanding model trained on a large text corpus. In this project we used the pretrained BERT model("bert-base-uncased"). We did not change BERT's parameters, and fine-tuned it with AdamOptimizer for 30 epochs [11].

5. Experiments, Results and Discussion

Evaluation Metrics

We will be looking at both the accuracy and the F_1 score as the metrics [1]2. For this classification task, we will define 'real news' as the positive class, and 'fake news' as the negative class. Given this definition, we naturally have the results of prediction in four categories: true positive (correctly predicting real news), true negative (correctly predicting fake news), false positive (incorrectly predicting real news), and false negative (incorrectly predicting fake news). The formula for accuracy is: Accuracy $= \frac{\text{true positive+true negative}}{\text{true positive+true negative+false positive+false negative}}$. For the F_1 score, we will calculate the means of precision and recall for both classes. The formula for precision is: Precision $= \frac{1}{2}\left[\frac{\text{true positive}}{\text{true positive+false positive}} + \frac{\text{true negative}}{\text{true negative+false negative}}\right]$. And the formula for recall is: Recall $= \frac{1}{2}\left[\frac{\text{true positive}}{\text{true positive+false negative}} + \frac{\text{true negative}}{\text{true negative+false positive}}\right]$. Using the result of the two formulas above, we get the F_1 score: $F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$ We will mainly be focusing on accuracy and F1 score for comparison of the models.

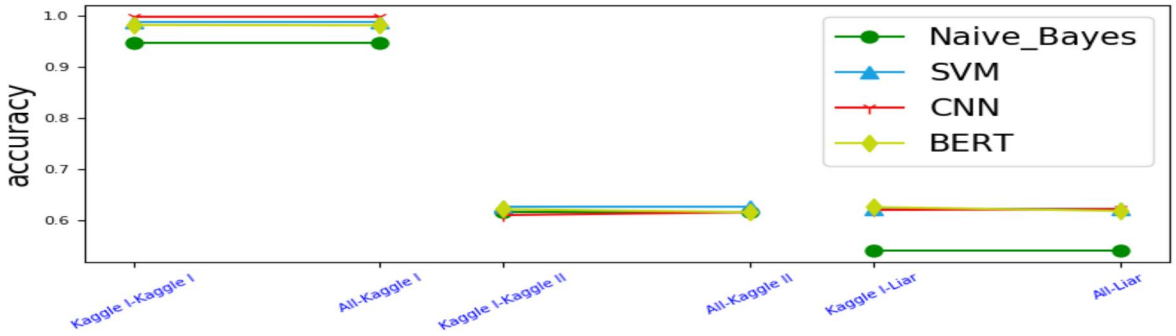

Figure 1: Figure 1
The Accuracy of four algorithms, trained with Kaggle I or all data, tested on Kaggle I, Kaggle III, and Liar test sets

Experiments and Results

Results for Naive Bayes, Support Vector Machine, CNN, and BERT

Table 1 2 contains the accuracy and F1 scores for the models on different test sets. For all models in Table 1, the training data is from Kaggle I. For all models in Table 2, the training data is from Kaggle I, Kaggle II, and Liar. The testing data is from Kaggle I, II and Liar respectively.

Accuracy/F1 score(fake)	Naive Bayes	SVM	CNN	BERT
Kaggle I	0.9473/0.9451	0.9880/0.9873	0.9989/0.9989	0.9824/0.9816
Kaggle II	0.6149/0.2796	0.6264/0.1558	0.6092/0.0286	0.6207/0.2917
Liar	0.5392/0.4184	0.6211/0.0000	0.6248/0.0096	0.6193/0.0881

Table1: Accuracy and F1 score of models trained on Kaggle I data

Accuracy/F1 score(fake)	Naive Bayes	SVM	CNN	BERT
Kaggle I	0.9473/0.9451	0.9880/0.9873	0.9987/0.9986	0.9816/0.9806
Kaggle II	0.6149/0.2796	0.6264/0.1558	0.6149/0.0290	0.6149/0.2472
Liar	0.5392/0.4184	0.6211/0.0000	0.6211/0.0095	0.6175/0.0367

Table2: Accuracy and F1 score of models trained on Kaggle I, Kaggle II, and Liar data

Precision (label 0/1)	Naive Bayes	SVM	CNN	BERT
Kaggle I	0.9396/0.9545	0.9915/0.9848	1.0000/0.9980	0.9756/0.9886
Kaggle II	0.4815/0.6395	0.5455/0.6319	0.2500/0.6176	0.4667/0.6389
Liar	0.4009/0.6366	0.0000/0.6211	1.0000/0.6223	0.5263/0.5263

Table 3: Precision of models trained on Kaggle I, Kaggle II, and Liar data

Recall (label 0/1)	Naive Bayes	SVM	CNN	BERT
Kaggle I	0.9507/0.9443	0.9832/0.9923	0.9978/1.0000	0.9877/0.9775
Kaggle II	0.1970/0.8704	0.0909/0.9537	0.0152/0.9722	0.2121/0.8519
Liar	0.4375/0.6012	0.0000/1.0000	0.0048/1.0000	0.0481/0.9736

Table 4: Recall of models trained on Kaggle I, Kaggle II, and Liar data sets.

Overall, all four types of models perform significantly better on Kaggle I test set. This is expected, since the majority of the training data came from Kaggle I. There is not much difference between the models

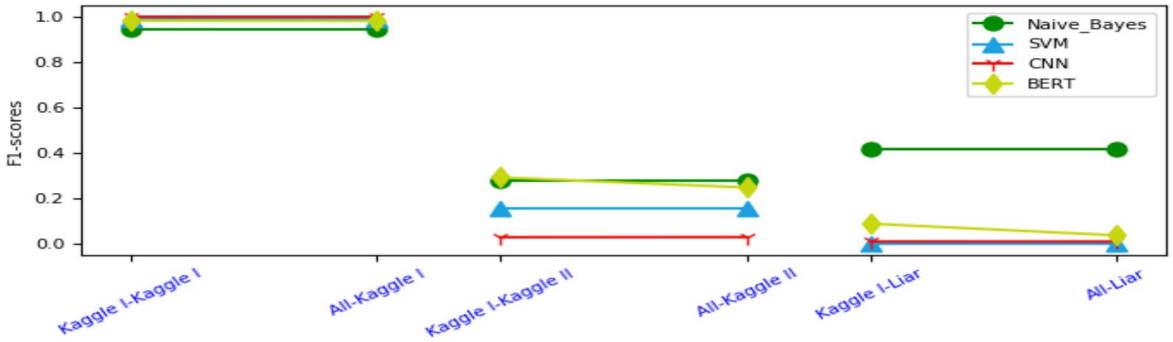

Figure 2: Figure 2
The F1 score fake class) of four algorithms, trained with Kaggle I or all data, tested on Kaggle I, Kaggle III, and Liar test sets

trained with Kaggle I data only and the models trained with all data. This might also be attributed to the unbalanced size among the datasets, as well as the homogeneity of Kaggle I data. The data in Kaggle I would have significantly higher frequencies of each word in the whole vocabulary. The news source of Kaggle I is also homogeneous, while all real news is from Reuters, and all fake news is from 21 Century Wire. In term of accuracy, the models trained with SVM, CNN, and BERT performs better than the Naive Bayes models(Table1,2, Figure1). In terms of F1 score, which reveals information about the accuracy within each class, the Naive Bayes models is significantly better than the other ones on Kaggle II and Liar test sets. Both CNN and SVM perform poorly on Kaggle II and Liar test set. From the precision and recall(Table 3 and 4), we can see that SVM does poorly when trained on all three datasets. The addition of Kaggle II and Liar has introduced more word to the full vocab, and thus more columns in the features, but not many data, since these are smaller data sets. This has caused disadvantage for SVM models.

Comparing bag-of-words features and tf-idf features

We also did parallel experiment on using bag-of-words features and df-idf features. We trained the model on Kaggle I data using Naive Bayes.

Accuracy/F1 score	Bag of words	tf-idf
Kaggle I	0.9473/0.9451	0.8262/0.7956
Kaggle II	0.6149/0.2796	0.3851/0.4513
Liar	0.5392/0.4184	0.4372/0.5339

Table 5: Accuracy and F1 score of models trained on Kaggle I data

	TP(Bag of words)	FP(Bag of words)	TP(tf-idf)	FP(tf-idf)
TN	1696	88	1265	519
FN	109	1847	131	1825

Table 6: Confusion matrix of models trained on Kaggle I data, and tested on Kaggle 1 test set

From the table above, we can see that the model trained on tf-idf features did not perform as well as the bag-of-words features. This is unexpected, since tf-idf should mitigate the influence of words that are frequent across the datasets, and also normalize on the length of each article. One possible cause might be that the model relies on words that are frequent across the datasets to classify the texts. For example, the word 'reuters' appear in many data labeled 'real news'. The tf-idf features might have affected the significance of this kind of word.

Top 30 Most Indicative Words for Fake News According to Kaggle I Training Data
'getty', '21wire', 'reilly', 'filessupport', 'bundy', '2017the', 'screenshot', 'acr', 'flickr', 'hilarious', 'henningsen', 'boiler', 'racists', '2016the', 'subscribing', '2017trump', 'finicum', 'antifa', 'wfb', '2017this', 'hannity', 'nyp', 'uninterruptible', 'wikimedia', 'lovable', 'spore', '2016featured', 'carlson', 'bullshit', '2017here'

Discussion of Qualitative Result: We get the top 30 most indicative words for fake news(above). It makes sense. For instance, "reilly" refers Bill O'Reilly, "hannity" refers to Sean Hannity, and "carlson" refers to Tucker Carlson, all of whom are conservative show hosts on Fox News.

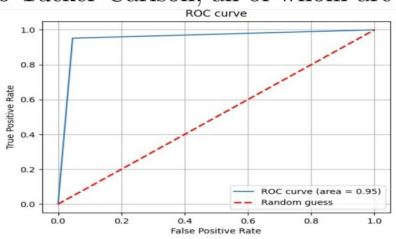

Figure 1: Trained with Kaggle I. Tested on Kaggle I. Use Naïve Bayes model.

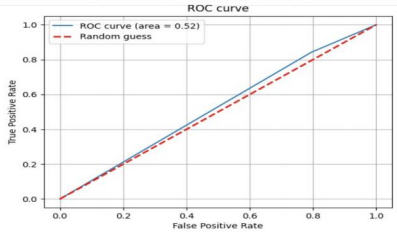

Figure 2: Trained with Kaggle I. Tested on Kaggle II. Use Naïve Bayes model

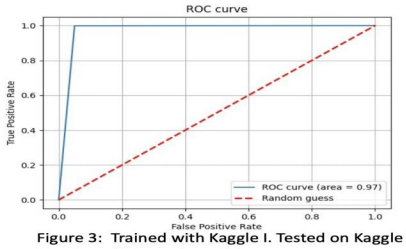

Figure 3: Trained with Kaggle I. Tested on Kaggle I. Use Neural Network model

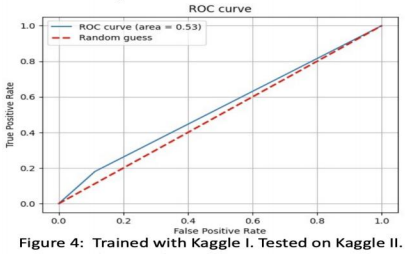

Figure 4: Trained with Kaggle I. Tested on Kaggle II. Use Neural Network model

Discussion of AUC/ROC: AUC is defined to be the area under the ROC curves (receiver operating characteristic). If we train and test using Kaggle I, AUC shows that our model outperforms random guess by a significant margin. But if we train using Kaggle I but test using Kaggle II, from the associated AUC, our model only performs slighly better than random guess. So there is an overfit issue here. The overfit issue is expected as Kaggle I only source from Reuters or 21 Century Wire, but Kaggle II uses other diverse sources.

The issue of overfitting and attempted solutions: Due to lack of diverse data source, overfitting has been a major issue in these experiments. The majority of the data are from Kaggle I, where all real news is from Reuters, while the fake news is from 21 Century Wires. In addition, a lot of data have the names of the source in the article, which leads to heavy bias. We tried to resolve the issue by switching the bag-of-words features with tf-idf features. We expected the new models to perform better on Kaggle II and Liar test sets. However, it failed to out-perform the model with bag-of-words features. We have discussed some possible causes above in the section where we compare these two features. In the future, we should implement the masking and dropout technique on the neural network to attempt at mitigating the issue.

6. Conclusion/Future Work

All four models perform equally well if Kaggle I is the test data, and equally poor if Kaggle II or Liar is the test data. This is because all the models overfit on Kaggle I. To improve on this issue and if we have more time, we should train the model with a lot more training data from a lot more diverse sources. This will reduce the overfit issues. We will also need to work on the masking and dropout techniques, in order to prevent bias caused by certain patterns in the text.

7. Contributions (Not Included In Page Limit)

Both partners have contributed equally to the project milestones. Yining Zhu works on the Support Vector Machine, Bert and sets up the overall structure of the project. Qianli Song works on the Naive Bayes and Convolutional Neural Network. Both partners have contributed equally to the clean-up of data and literature reviews.

8. References (Not Included In Page Limit)

1. Detecting Fake News with Machine Learning Method, by Supanya Aphiwongsophon and Prahbas Chongstitvatana. Supanya Aphiwongsophon and Prahbas Chongstitvatana are Faculty of Computer Engineering at Chulalongkorn University, Thailand.The paper is published in 2018 during the 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology by IEEE Explore. 18-21 July 2018 https://ieeexplore.ieee.org/abstract/document/8620051.

2. Detecting Fake News using Machine Learning:A Systematic Literature Review, by Alim Al Ayub Ahmed, Ayman Aljarbouh, Praveen Kumar Donepudi, Myung Suh Choi. Alim Al Ayub Ahmed is from School of Accounting, Jiujiang University, China. Ayman Aljarbouh is from Department of Computer Science, University of Central Asia, KYRGYZSTAN. Praveen Kumar Donepudi from Enterprise Architect, Information Technology, UST-Global USA. Myung Suh Choi is from Monta Vista High School, Cupertino, CA. The paper is published on arXivLabs, 8 Feb 2021. https://arxiv.org/abs/2102.04458.

3. Supervised Learning for Fake News Detection. Julio C. S. Reis, Andre Correia, Fabrıcio Murai, Adriano Veloso, and Fabrıcio Benevenuto. All authors are from Universidade Federal de Minas Gerais. The paper is published in IEEE Intelligent Systems (Volume: 34, Issue: 2). March-April 2019. https://ieeexplore.ieee.org/abstract/document/8709925.

4. Which machine learning paradigm for fake news detection. Dimitrios Katsaros, George Stavropoulos and Dimitrios Papakostas. Dimitrios Katsaros is from University of Cyprus, Nicosia, Cyprus. George Stavropoulos is from University of Thessaly, Volos, Greece. Dimitrios Papakostas is from University of Thessaly, Volos, Greece. The paper is published 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI). The publisher is IEEE. 14-17 Oct. 2019. https://ieeexplore.ieee.org/abstract/document/8909583.

5. A Benchmark Study on Machine Learning Methods for Fake News Detection. Junaed Younus Khan, Tawkat Islam Khondaker, Anindya Iqbal and Sadia Afroz. Junaed Younus Khan, Tawkat Islam Khondaker and Anindya Iqbal are from Department of Computer Science and Engineering, Bangladesh University of Engineering. Sadia Afroz is from Technology International Computer Science Institute. The paper is published on ResearchGate.net. 14 May, 2019. https://www.researchgate.net/profile/Md-Tawkat-Islam-Khondaker/publication/333077208_A_Benchmark_Study_on_Machine_Learning_Methods_for_Fake_News_Detection/links/5d1198bea6fdcc2462a35118/A-Benchmark-Study-on-Machine-Learning-Methods-for-Fake-News-Detection.pdf

6. Fake and Real News dataset.Clément Bisaillon. 2020. https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset?select=Fake.csv.

7. Source based Fake News Classification. Ruchi Bhatia. 2020. https://www.kaggle.com/ruchi798/source-based-news-classification.

8. LIAR: A Benchmark Dataset For Fake News Detection. William Yang Wang. 2017. https://github.com/thiagorainmaker77/liar_dataset

9. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

10. Definition of Precision = True Positive/(True Positive + False Positive). Definition of Recall = True Positive/(True Positive + False Negative). Definition of Accuracy= (True Positive + True Negative)/Total.