CS 235 - Data Challenge

Kelsey Musolf
kmuso001@ucr.edu
University of California Riverside
Riverside, CA, USA

ABSTRACT

With the new advancement in AI tools open to the public, many have started to question the creators of text, images, and videos; was it made by a human or AI? This project's goal is to train a model that can do just that for text. Given some text, was the creator of it human or AI? I will use models such as logistic regression, SVM, BERT and RoBERTa to find the best classifier while also trying and testing different parameters. The end of my project will have a comparison between the different models and methods I tried, and have a working model that performs the best.

KEYWORDS

logistic regression, SVM, Random Forest, BERT, RoBERTa, SVD, TF-IDF

1 INTRODUCTION

This project utilizes the data set RAID, famously known for the wide range of AI models used within the dataset. RAID is used for AI text detection, with approximately 11 GB of entries spanning from human written text to AI written text by 'llama-chat', 'mpt', 'gpt2', 'mistral', and 'cohere'. I have been tasked with comparing models to detect human or AI text using techniques learned in class during lecture and while working on the methods project.

2 METHODOLOGY

I made three models and tested them against BERT. Logistic regression, SVM, and Random Forest were all run with TF-IDF and SVD for dimension reduction. I ran each model 4 times (totaling in 12 models total). I started with the most simple, TF-IDF parameter set to (1,2) words per unit, and SVD = 100. I ran this on all three models (LR, SVM, RF). I then changed one parameter: TF-IDF now has (1,3) words per unit to look at. I ran again, changing TF-IDF back to (1,2) and increased SVD to = 500 features. And finally, I combined the two to be TF-IDF (1,3) and SVD = 500. The best model results was SVM with TF-IDF = (1,2) and SVD = 500 features.

I took the parameters that worked best on average for all models, and decided to use these parameters to test against my hold out set. These parameters were TF-IDF = (1,2) words per unit, and SVD = 100. Testing against the holdout set, however, showed that logistic regression performed the best here. In the end I compare BERT to this logistic regression model result on the hold out set.

When comparing models, I used the area under the ROC curve. This represents how well each model truly works, not just accuracy where false positives are added to the score and false negatives are left out, more accurately, the ROC curve is the true positive rate, AKA sensitivity, against the false positive rate which is 1-specificity (Lecture slides, 05c-supervised). The higher the score, the better the result is.

Throughout the project, I had to make many decisions. A few major ones include the following:

2.1 Sampling

Due to the large size of the RAID dataset, I've had to adjust by taking a smaller sample. I decided to take 10,000 records. A few things I did to help my models, I ensured this had a 50 50 split between AI and Human text, I used random state = 42 to keep it reproducible and that every time I ran my code this, I got the same 10k sample, and the sampling was uniform random sampling.
Note: The sklearn library automatically has a way to split the data in a fair manner, keeping the AI and Human text equal to take into account any class imbalance. It is sampling without replacement so there is no overlap between each fold.

2.2 Train and Test Split

For my more basic models such as Logistic regression, SVM, and Random forest, I used stratified k fold for training and testing. I used 5 folds because k is not too large nor too small, it is also what I used in the methods project.
I also decided to take out 10% before splitting into train and test that I use later in my project for final testing; my holdout set.

2.3 Text to Vector

After doing the train and test split, I chose to use TF-IDF for the text to vector. Because of this, I will also be using SVD to reduce the number of dimensions. I chose TF-IDF because it is a simple concept where words are represented as numbers based on their importance (frequency plus inverse document frequency - rarity of a word for all text 'documents') and are reflected in a vector. I can change the parameters for this technique, taking different unit sizes. These units look at words or groups of words and the number of words per unit is labeled as ngram. Initially it is set to ngram = (1,2) (units on a single word and pairs of words) and then change it to ngram = (1,3).

2.4 Models chosen

I chose to train on Logistic Regression, SVM, and Random Forest. I wanted to try different parameters for TF-IDF and SVD to see what model would be best and use that as my baseline. It ended up being SVM with the parameters of TF-IDF ngram= (1,2) and SVD max features = 500 until I tested on the holdout set, in which case logistic regression performed the best with parameters TF-IDF ngram= (1,3) and SVD max features = 100.

2.5 BERT

I chose to run BERT on the 10,000 sample (same as simpler models). This model takes multiple epochs of the data and uses these

to calculate the correct weight. However, due to time, I only ran one epoch on my BERT model. This is counter productive to the performance of the model.

Because BERT works better the more epochs it has, I trained a BERT model on a sample of size 1000 about 5 times. This model works best with more training by design, so even with a smaller sample size, the increase in epochs will improve the model when tested against the hold out set.

3 RESULTS

The results of my models and testing are best shown with the following graphs. Figure one shows the results of my models logistic regression, SVM, and random forest, with the parameter variations. Figure two shows the results of the basic models when tested on the holdout set of data. Figure three shows the best model against both BERT tests (big and small). Lastly figure four shows the models with various parameter changes tested on holdout set against the two BERT model sizes, graphing all models from my data challenge to the same plot.

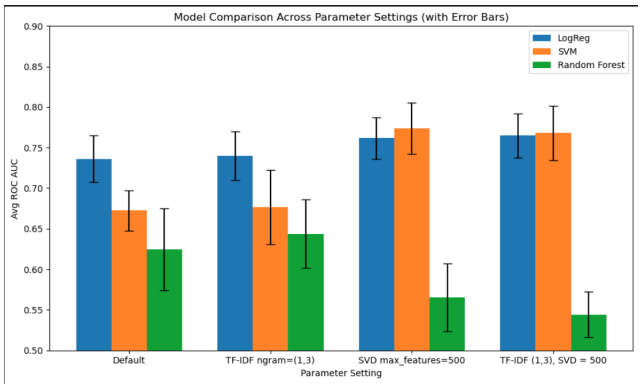

Figure 1: Models tested with different parameters

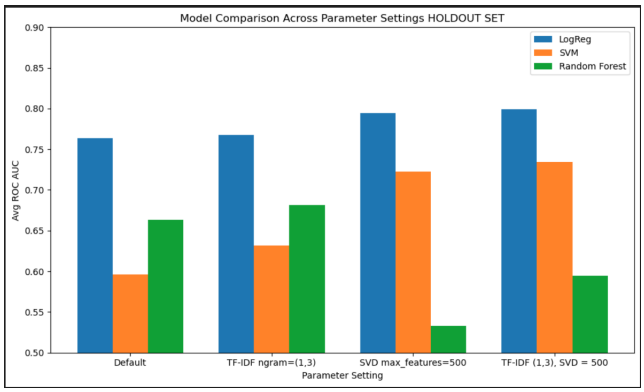

Figure 2: Models tested on holdout set with different parameters

Comparing the results from training versus testing onthe hold out set it is apparent that logistic regression on average will perform the best. However, SVM model's performance drastically drops when testing on the holdout set. This is why it is important to test on the holdout set, with out this plot SVM would be considered to be just as good as logistic regression for some parameter settings. However, the holdout set proves this to not be true, the training done was perhaps overfit and relayed much better scores.

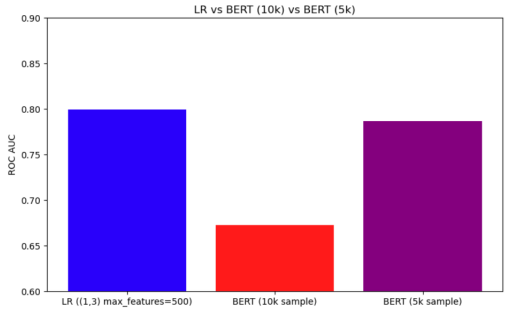

Figure 3: LR compared to both BERT models

The different sizes of data for the BERT model did show a difference, but this is because of the number of times I trained each size model. Trained once, the 10k sample for BERT performed very poorly. It did not have time to keep readjusting the weights of the model when given only one epoch. Meanwhile, the 5k sample did great with 5 different epochs for training.

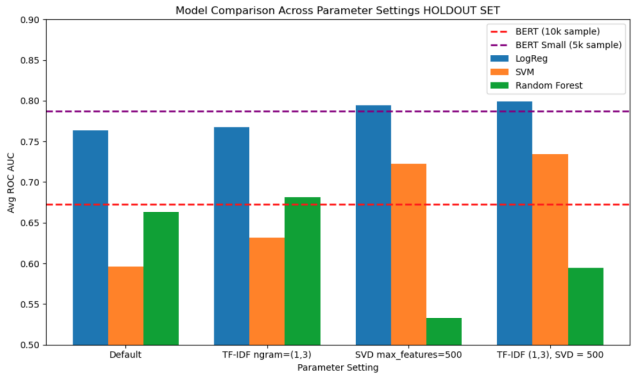

Figure 4: All models compared to all BERT models

Figure 4 is by far the most interesting graph of my project. It shows the test against the holdout set for every model made in my project. My prediction: if the 10k BERT sample had 5 epochs, it would perform the best, surpassing my logistic regression model and outperforming every other model tested in my data challenge.

If requested to detect AI written text versus human text, I would use my logistic regression model. Given more time, I would train my BERT model more and use that, but with my current results, logistic regression will be the best option.

4 CONCLUSION

I was able to successfully train and test multiple models on detecting human text versus AI written text. I used my knowledge from class CS235 Data Mining taught by Vagelis Papalexakis and the skills I learned from the class Methods project to make models that performed well enough to compare to a model such as BERT.

5 FUTURE WORK

Given more time and perhaps a better computer to run my tests on, I would continue to expand the size of the sample for BERT and

largely increase the number of runs and epochs used for training. I would also expand my project to add RoBERTa as listed in the paper [1].

6 REFERENCES

[1] Liam Dugan, Andrew Zhu, Firoj Alam, Preslav Nakov, Marianna Apidianaki, and Chris Callison-Burch. 2024. *GenAI Content Detection Task 3: Cross-Domain Machine-Generated Text Detection Challenge*. arXiv preprint arXiv:2405.07940. https://arxiv.org/abs/2405.07940