COMP 550: PROGRAMMING ASSIGNMENT 1

CALEB MOSES*

1 INTRODUCTION

This report describes a text classifier trained on fake and real facts about New Zealand birds generated by a Large Language Model, Llama [1] by Meta AI Research.

2 DATASET GENERATION

I began by taking the open source model Llama-2 and prompting it to produce bird facts using the following prompts:

Prompt 1 Real bird facts

```
SYSTEM: You will generate {num_real_facts} real facts about the {bird_name}
    bird. Each fact should be one to two sentences long. You must number each
    line, and format each response starting with the tag [START] and ending
    with [END]. For example: '1. [START] The {bird_name} is native to XYZ
    region. [END]'
USER: Give me {num_real_facts} facts about {bird_name}.
```

Prompt 2 Fake bird facts

```
SYSTEM: You will generate {num_fake_facts} fake facts about the {bird_name}
    bird. Each fact should be one to two sentences long. You must number each
    line, and format each response starting with the tag [START] and ending
    with [END]. For example: '1. [START] The {bird_name} can speak three
    languages. [END]'

USER: Give me {num_fake_facts} facts about {bird_name}.
```

The number of real and fake facts were each set to 50, and the bird name was interpolated across the following ten New Zealand birds:

- Bellbird (Korimako)
- Fantail (Pīwakawaka)
- Kea bird
- Kererū
- Kiwi bird

* *PhD Student, School of Computer Science, McGill University, Montreal, Canada*

- New Zealand Falcon (Kārearea)
- Pūkeko
- Rock Wren (Pīwauwau)
- Tomtit (Miromiro)
- Tūī

This resulted in 10 birds times 50 prompts each with real and fake facts, totalling in 10 x 50 x 2 = 1000 unique facts.

3 EXPERIMENTAL PROCEDURE

Once I had generated the full list of facts to work with, I separated them into a train-dev-test split according to a 60-20-20 rule. This means my training set contained 600 samples, and my dev and test set each contained 200.

3.1 The range of parameters

I then wrote a model factory to allow me to initialise a wide range of models each with different hyperparameters and different pre-processing, covering the following:

- Lemmatization
- Stop-word removal
- n-grams
- Stemming
- Treebank Part of Speech Labelling
- Vocab limit
- Classifier type

For the classifier type, we considered the SGD classifier, Logistic Regression, Naive Bayes and Support Vector Machine. To make things easier, each classifier was implemented according to its defaults in Scikit-Learn. N-grams were considered in the range of 0-5, and the stop-word list, lemmatization, stemming and so forth were implemented using nltk.

3.2 Results

In order to determine which parameters were the most effective for improving the model, a logistic regression was implemented. We assumed the model accuracy was a linear function, including an intercept and parameters for each of the model hyperparameters, including the classifier type. We then trained 1000 different models according to randomly selected hyperparameters and used these to fit the logistic regression model. The resulting coefficients of this model are given in Table 1

From the regression we can see that on average most of the models did quite well since the Intercept is 0.975. From there, the choice of classifier seemed to make the most significant positive difference in the model, with SVM and logistic regression doing quite well. From there, considerations for pre-processing and so forth were of secondary significance.

Feature	Coefficient
Intercept	0.975430
SVM	0.006330
Logistic	0.006153
SGD	0.003856
Vocab Limit	0.003698
N-grams	0.000948
Lemmatization	0.000368
Treebank POS	0.000076
Stemming	-0.001314
Stopword Removal	-0.001812
Naive Bayes	-0.016053

Table 1: Coefficients from the logistic regression model representing the influence of each hyperparameter on model accuracy.

3.3 Conclusion

We were able to show that training a linear classifier with reasonably high accuracy (0.975% on average across 1000 trials) is quite straightforward. It was not clear to me before we did this that fitting a classifier with high accuracy on a task like this would be possible, so that was interesting.

3.4 Limitation

The idea that word frequencies correspond to truth/falsehood is a simplifying assumption that we made in order to implement this model. The truth is word frequencies are not sufficient to estimate truth/falsehood, and in my own experiments I was able to generate many examples that could fool the model. For that reason I think it would be more accurate to say that we trained a model that was fit for distinguish some generated text fitting a fairly tight set of criteria.

REFERENCES

[1] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.