COMP90051 Statistical Machine Learning Project1

Jiayi Xu 1165986 Runyu Yang 1118665 Zhenghan Zhang 1136448

1. Introduction

This report discusses the approach followed to build a better performance classification model to recognize the different text written by human or machine. Two datasets (set1 & set2) used are consisting of different texts written by human or machine based on prompts. For evaluating classification model, datasets are split into a training set and a validation set, performances of models are evaluated by comparing the output of validation set. This report mentions several different models performance, including a Logistic Regression Model (Baseline Model), five Gating Recurrent Unit (GRU) models and a Bert model.

2. Data Preprocessing

In this stage, datasets are processed in three steps. To do data split, the validation set is formed by randomly selecting 200 human labels and 200 machine labels, and the rest of the set1 is used as a training set. Since the set2 is much smaller than set1, number of selections for different labels changes to 20. By inspection of distribution (Figure 1), the words which have frequencies below than two have a smaller influence on relationship between prompt and txt. The relevance between these words and the texts is not considered in models. Lengths of the prompt and txt of raw data are recorded before processing the low-frequency words. Figure 2, prompt and text length distribution, reveals that the machine-generated text is longer than the human-written text.

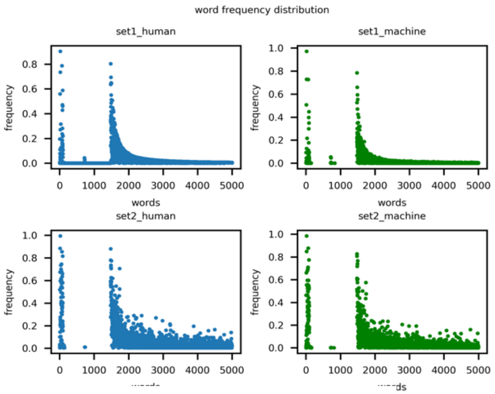

Figure 1 Word Frequency

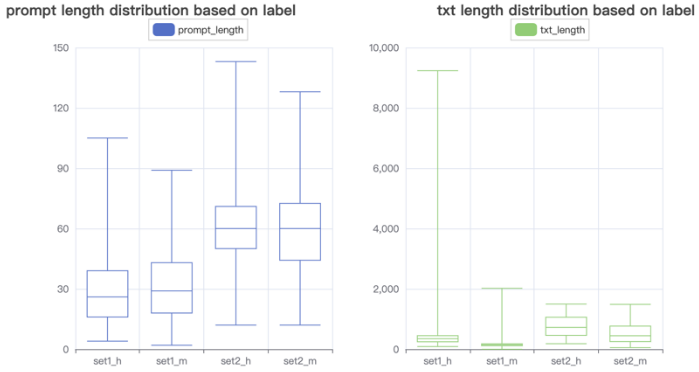

Figure 2 prompt and txt length distribution

3. Selected Models

This dataset is mainly processing information in sequence of words. From preprocessing part, known that sequence length is long and with different text length, which cannot use common machine learning way to deal with. To keep sequential information, using recurrent neural network (RNN) should be a good attempt. Prediction models with majority algorithm reaches the best performance.

3.1 Gating Recurrent Units (GRUs)

Naive RNN could not handle very long sequence by gradient descent, GRU is chosen to use. The application of GRU not only allows important information to be extracted and long-term information to be retained effectively, but also train fast within RNN family (Chung et al., 2014). Txt is generated based on prompt, by thinking about similar task, translation, is also about two sequential texts. Sequence-two-sequence(seq2seq) structure for translation, used to present the relationship from prompt to txt, in this case, sequence data are mapped to the longest text length within batch use symbol *'<pad>'* (Sutskever et al., 2014).

Simple Gating Recurrent Unit: Modelling embedding layer and GRU layer for txt and prompt separately. Final hidden state of prompt GRU layer is used as initial state of txt GRU. Txt GRU outputs in each step represent information explained by current words. The *sequence_mask* function clears irrelevant items to zero, results of correlation part are sum of information usefulness of prompt and txt. Furthermore, for classification, two fully connected layers and a *SoftMax* function map inputs to range [0,1] for training and prediction.

Two GRU with attention models: In simple GRU model, the last hidden state may not be the most useful information for current state. Therefore, this problem is solved by introducing an additive attention mechanism which weighting of all states. (Quinn et al., 2020).

Attention mechanisms with parameter None: By passing the parameter None, all data in prompt is weighted for attention calculation and applied to output generation of txt GRU layer.

Attention mechanisms with parameter prompt_length: A weighted attention calculation is performed on lengths of prompt data and applied to outputs of txt GRU layer.

Bidirectional GRU Model with Attention: Thinking about that prompt also can be summarized from txt. Correlations between prompt to txt and txt to prompt under attention mechanism are merged and applied to txt GRU layer, the fully connected layer with double size of input.

GRU Longer Model: In this model, due to lengths of set2 machine txt are around double longer than that of set1 machine txt on average. A 50% probability of randomly splicing text data with the same machine ID or human when training domain1, it allows GRU model could performance better on longer.

3.1.1 Parameters Applied in Model

For all models mentioned above applying the same model parameters: vocab_size is the size of vocab after deleting low-frequency word, embedding size of prompt is 64 and embedding size of txt is 128. There are two rnn layers and neurons in rnn layers are 16. Moreover, symbol <pad> used to populate the missing value should be added into the vocab dictionary.

3.1.2 GRUs Performance

Depend on comparing accuracy and macro_f1_score of two models, the overall performance of GRU with attention (param = None) model is better than simple GRU model both on validation sets and training sets of set1 and set2. By analyzing graph of accuracy, macro_f1_score and train loss of validation set of GRU with attention (param = prompt_length) model (Figure 3), the line of accuracy jitters up slowly and loss decreases sharply before epoch5 and then trends to flat. What is more, it is found that model tends to overfit after epoch 5 when comparing classification performance. The same occurred for directional GRU with attention model.

Table2: Accuracy and Macro_f1 of Model

	Simple GRU Model			
	Set1		Set2	
	Train	Val	Train	Val
Accuracy	0.94	0.90	0.56	0.62
Macro_f1_score	0.94	0.90	0.45	0.56

	GRU Model with Attention							
	Parameter = None				Parameter = prompt_length			
	Set1		Set2		Set1		Set2	
	Train	Val	Train	Val	Train	Val	Train	Val
Accuracy	0.96	0.96	0.68	0.60	0.95	0.96	0.63	0.65
Macro_f1_score	0.96	0.96	0.65	0.52	0.96	0.96	0.53	0.53

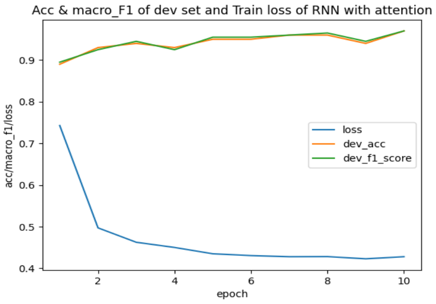

Figure3 Acc&Macro_F1&Train loss of RNN with attention

As a result, parameters of pre-trained model epoch 7 are finally used to fine-tune with set2 data. Accuracy of set2 validation set is improved to 0.72 by reducing learning rate and weights of pre-train model, the fine-tuned model is also at risk of overfitting. Performances of the bidirectional and GRU longer models on validation set are not better than that of attention models. Due to relative complexity of the models, training is stopped in first 3 epochs to reduce the risk of overfitting. Therefore, a prediction model with majority algorithm is constructed by combining individual models with similar performance as described above.

3.2 Use Majority Algorithm for Prediction

Four GRU models discussed above all achieved accuracy least of 0.9 on set1 and fine-tuning on set2 reaches different level of accuracy. Based on above analysis of GRU models with attention has risk of overfitting in the classification, thus fine-tuned model based on this model and set2 also has the risk of overfitting.

Table3: Accuracy Table of GRU Model

	Accuracy
Simple GRU	0.744
GRU with Attention (None)	0.756
GRU with Attention (prompt_length)	0.738
Bidirectional GRU with attention(None)	0.724
GRU longer	0.744
GRU prediction with Majority	0.794

GRU prediction model with majority algorithm achieves the best performance on Kaggle (0.794). This is because majority prediction model not only preserves important information processing mode of GRU to sequence data, but also reduces the risk of overfitting of model to predicted value. Nevertheless, majority algorithm also has disadvantages. Majority algorithm is only suitable for model with similar performance, otherwise there is a risk of errors in the final majority element determination.

4. Other Attempted Models

4.1 Logistic Regression Model

Logistic Regression as a baseline model, because it used tf-idf, in this approach, can not get the sequentional information within a sentences. Classification report of validation set of set1 and set2 shows that accuracy and F1_score of set1 are both 0.86, and those of set2 are 0.57 which will be used as baseline to evaluate (Table 1).

Table1. Classification Report of Validation set

	Set1				Set2			
	Precision	Recall	F1_Score	Support	Precision	Recall	F1_Score	Support
0	0.85	0.89	0.87	200	0.57	0.65	0.60	20
1	0.88	0.84	0.86	200	0.59	0.50	0.54	20
Accuracy			0.86	400			0.57	40
Macro avg	0.87	0.86	0.86	400	0.58	0.57	0.57	40
Weighted avg	0.87	0.86	0.86	400	0.58	0.57	0.57	40

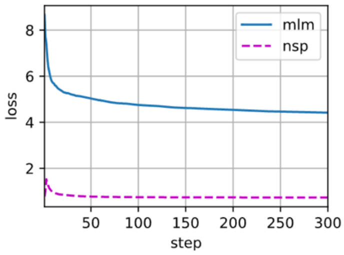

Figure4: Train loss of pre-trained model

4.2 Bert Model

By thinking of bert structure generally performance better than rnn models in real tasks, it used transformer to include neighbering words information which might give a highrt performance. Since all the text data in this project have been converted into digital format, a pre-trained Bert model is built from scratch based on only set1 data. Training data is merged with prompt and txt, at beginning and end of the sentence adding symbol <CLS> to separate different sentences and symbol <SEP> between prompt and txt messages. Three different embeddings are used for data to store word, segment, and position information of one sentence. The Masked LM function is used to mask the text in pre-trained model, using an 80%:10%:10% ratio for word masking (Devlin et al., 2019). For Next Sentence Pair Prediction, 50% :50% ratio applied pairs randomly replacement. Disappointed that performance of Bert model does not meet the standards of the baseline model. These three reasons lead to the failure of the pre-trained Bert model to fit and converge on the set1 (Figure 4). Firstly, the amount of training data does not reach the amount that can make pre-trained model converge. Secondly, Bert model takes very long time to train until the model converges. In addition, open-source pre-trained bert model can not use for digital sequences.

5. Fine-tuning on Domain2

If directly applied the model trained from set1 on set2, it does not perform well. The reason is that the distribution of set2(human: machine = 1:4) does not match that of set1(human: machine =35:1). Thus, model trained based on set1 is used as a pre-trained model and its parameters are fine-tuned with set2 data. Fine-tuned model can learn machine and human textual information from set2 and then make prediction. Fine-tuned model only used on set2 retains accuracy of set1 model while improving accuracy of binary classification predictions on Domain2 data.

In terms of parameter tuning for domain2, five individual GRU models performed differently. Accuracy of prediction model with majority algorithm improved to 0.8 on the validation set of domain2 when compared to individual models. This is due to the fact that model can select predictions that occur more frequently based on majority algorithm. Depend on parameters tuning for domain2, accuracy of predictions for test data reaches the current best of 0.794.

6. Conclusion

In conclusion, GRU prediction model with majority algorithm performs best in this project. If txts have not been digitized with more training data and time, Bert model will have a better performance than GRU prediction model based on association between txts and prompts. Moreover, lengths of txt are not processed much which could be manipulated in the future to achieve better model performance and prediction results.

7. Reference

Chung, J., Gulcehre, C., Cho, K. H., & Bengio, Y. (2014, December 11). *Empirical evaluation of gated recurrent neural networks on sequence modeling.* arXiv.org. Retrieved April 26, 2023, from https://arxiv.org/abs/1412.3555.

Devlin, J., Chang, M.-W., Lee, K., &amp; Toutanova, K. (2019, May 24). *Bert: Pre-training of deep bidirectional Transformers for language understanding.* Retrieved April 25, 2023, from https://arxiv.org/abs/1810.04805.

Quinn, J., McEachen, J., Fullan, M., Gardner, M., & Drummy, M. (2020). *Dive into deep learning: Tools for engagement.* Corwin, a SAGE Company.

Sutskever, I., Vinyals, O., &amp; Le, Q. V. (2014, December 14). *Sequence to sequence learning with Neural Networks*. arXiv.org. Retrieved April 27, 2023, from https://arxiv.org/abs/1409.3215.