Authorship Attribution for Neural Text Generation

Venkatanand Sai Duggirala
Sarwabowma Sri Sai Karthikeya Sarraju
Harinath Reddy Cingapuram
Sai Lakshmi Tejaswini Kallakuri
University at Buffalo
Buffalo, New York
{vduggira, sarwabow, hcingapu, kallakur}@buffalo.edu

Abstract

There are tremendous advancements in Natural Language Processing, such as the OpenAI chatbot models. These technologies could be used in many ways to help the society but simultaneously could pose certain risks either with or without guilt. Thus in this project we would analyze some of the research done till date in order to solve the problem. Three problems would be analyzed, that is if two texts are generated by the same NLG, if text is from a human, and classify the texts with respect to the NLG. We consider some of the Neural Language Generation (NLG) models such as GPT-3 and try researching different Linguistic Features such as POS statistics, stylometric features and so on. We used two different BiLSTM architectures and two kinds of tokenizers as well as compared the results with that of the Random Forests. The data is generated from 10 NLG methods and a human. In conjunction to this we also carried out a Reddit case study where we extracted the data from 2 NLG methods and a human and experimented with the models that we used on the actual set and analyzed the results.

Keywords: Natural Language Processing, Neural Language Generation, Linguistic Features.

1 Introduction

With the developments in the field of chatbots which could mimic the human language, there's a high ground for the models to be used as part of the deep fakes. Thus in this project we have analyzed various prior work in order to solve the following three questions.

1. Same or not: Whether the generated text is from the same NLG method (human) or not.

2. Human vs bot: Whether the text is from a human or an NLG method.

3. Which NLG method: Whether the text is from the ith NLG method from a set of k methods.

The above questions are typical binary or multi label classification problems. Hence the project would deal with the domain of the classification of text using different Machine Learning, Deep Learning models such as the RNN and so on.

Five models were built for the first problem and four models for the second and third problems. The two kinds of Bidirectional Long Short Term Memory (BiLSTM) model architecture was used for the four models and an XG Boast was used for task 1. In conjunction to this the tokenization methods used were of two types. One is Natural Language ToolKit (NLTK) package tokenizer and another one is the Bidirectional Encoder Representations from Transformers (BERT) tokenizer.

Experimentation was carried with removing the punctuation, stopwords as well as without removing them. While using stylometric features, the stopwords as well as the punctuations were not removed in order preserve the syntactic style of the text.

The datasets have been collected from the Authorship Attribution repository[1]. Along with that the data was also generated from two NLG methods and taken into account. A Reddit case study was also carried out with the models on all the three tasks. A detailed error analysis has been carried out and also the results of task 3 were compared with that of the three Random Forests models either with Attention mechanism or without it. The BERT tokenizer was also used in conjunction with the Random Forests model.

2 Related Work

Different papers were reviewed. In the paper Experiments with Convolutional Neural Networks (CNN) for Multi-Label Authorship Attribution[2], A multi layer CNN model that computes probability distribution if a single label task or the average of the probability distributions in case of a multi label task. To control the overfitting the batch normalization technique has been used. The max-pooling method used to keep the maximum value across the sentence. The ELU activation function used in the hidden layers and the softmax / sigmoid activation at the output layer. The sentences are represented using word2vec and glove embeddings. The macro F1 score is 0.736, micro F1 score is 0.744 and the accuracy is 65.3 . In the paper for Style-aware Neural Model with Application in Authorship Attribution[3], Two models. The first one is the Lexical and Syntactic encoding model where the pre-trained Glove embeddings are used for lexical and POS tags for the syntactic purpose. The second one is the Hierarchical model where the inputs from the first model are taken in by two identical CNN models which use temporal max-pooling. Then the output is fed to a sentence level encoder which gives the semantic/syntactic representation of the document. Both representations are fused. Softmax activation function is used for classifying. The results were 90.58 on CCAT10, 82.35 on CCAT50, 72.83 on BLOGS10 and 61.19 on BLOGS50.

In Exploring syntactic and semantic features for authorship attribution[8], A Multi-Channel Self Attention Network (MCSAN) has been used which has four channels i.e., Word, POS, Phrase and Dependency channels. These features are then combined and fed to a multi-channel self attention model which extracts the contextual information and this is fed to a CNN model with max-pooling layer combined with an Long Short Term Memory (LSTM) model with softmax activation function for the classification. Experimentation has been carried out with BiLSTM and TextCNN also. The results were 92.89 on CCAT10, 83.42 on CCAT50 and 89.97 on IMDB62. In the paper, Leveraging Discourse Information Effectively for Authorship Attribution[10] The CNN2PV model has GR (Grammatical Relations) and RST discourse relations probabilities. The CNN2DE has the discourse embeddings where CNN2 is the baseline model and it uses an embedding layer, a convolution layer, max-pooling and the softmax activation function for the purpose of classification. The Rhetorical STyle(RST) model performs better than GR as it leverages the weight learning in CNN and also provides more fine-grained features. The results were 98.8 on novel - 50 and 92.0 on IMDB62.

In the paper, an evaluation of authorship attribution using random forests[12], Random forests are known to handle noisy data, frequent characters, frequent ngrams, freq wordlen, freq rewriterules, freq wordshapes have been used since decision trees could handle numerical data well compared to the nominal data. The task is divided into three problems namely A with 3 classes, C with 8 classes but bigger size compared to A and I with 14 classes and much bigger size. At first the features for a given document along with its label is derived and used with WEKA to build a classifier. The results were 100 on Prob A, 87.5 on C and 92.9 on I.

In the paper, Authorship Attribution vs. Adversarial Authorship[13] from a LIWC and Sentiment Analysis Perspective, Preprocessing the data by getting the tf-idf values and standardizing as well as normalizing them to get unit feature vectors. LIWC provides insight from a psychological point of view. All the 93 output variables are used as the features. These provide word frequencies. Three different models Linear Support Vector Machines(SVM), RBF SVM and Multi-Layer Perceptron(MLP) have been used. Stratified 4 fold cross validation was implemented to get good accuracy. The results were 78 on MLP, 71 on RBFSVM and 84 on LSVM. In the paper, Topic or Style?[14] Exploring the Most Useful Features for Authorship Attribution, Three kinds of linguistic features have been selected. They are style, content and both. The style-based are punctuation, digits, words and so on. The content-based are bags of n-grams representing the topical words. The style-based have 174 words and 12 punctuation. Both character and word n-grams are limited to bi and tri. Two models were used i.e., the single Feed-Forward Network(FNN) and Logistic Regression(LR). The style features has two sub groups i.e., lexical and syntactic. For FNN softmax activation function has been used. The results were Style: on Judgment 91.07, on CCAT10 76.00, on CCAT50 72.72, on IMDB62 95.93; Content: on Judgment 91.51, on CCAT10 76.20, on CCAT50 72.88, on IMDB62 95.59; Hybrid: on Judgment 91.21, on CCAT10 74.80, on CCAT50 71.76, on IMDB62 95.26.

Thus one could conclude that there is a lot of research that has been put into the Author Attribution field. There are various approaches such as the stylometric approaches i.e., lexical or syntactic models or using content based n-gram methods. One could use the stylometric aspect of the topic like in syntactic models. One could also take into account the discourse of the topic such as the RST model or combine all of the above features. Also one could even use different Machine learning or Deep learning models such as the RNN. Thus with the experimentation and research, the accuracy with which the prediction could be done, can be achieved.

3 Methods/Model Architecture

3.1 Datasets

We have extracted the data using OpenAI key for the InstructGPT and GPT3 models. For the rest of the models and the human the data was taken from the GitHUB repository[1]. The prompts used to generate the text from GPT3 and InstructGPT are the same as those used for the rest of the NLG methods. Each of the tasks have been considered separately hence the datasets were also prepared separately. The data is taken from the 10 NLG methods (CTRL, GPT, GPT2, GPT3, Instruct GPT, GROVER, XLM, XLNET, PPLM, FAIR) and 1 human. For task 1 which is a binary classification problem, the dataset consists of two columns chat1 and chat2; and also a third label column with two classes True and False. Two sets have been prepared, one for balanced (1:1) and another one for imbalance (1:10)[1]. For task 2, which is also a binary classification problem, the two sets were made, where one is balanced (1:1) and another imbalance (1:10)[1].Here for the task 2 there's only a single column for the chat generated and another column for the label which has two classes, True for human and False for the chats generated by all the NLG methods. For task 3 we have prepared a large dataset consisting of all the NLG methods and a human along with labels corresponding to each generator[1]. There's a single column for the chat and a column for the label.

For the Reddit case study, prompts were extracted using the PRAW method and using the different topic names. We have data from 2 NLG methods (GPT3, InstructGPT) and a human. For task 1 the dataset has 2 chat columns and one label column. For this task two sets, one balanced set (1:1) and another imbalanced set (1:2) were made. For task 2 the dataset has one chat column and one binary label column. The dataset is split into 2 sets one for balanced (1:1) and another for imbalanced (1:2). For task 3 we have 3 labels with a single chat column. The data from NLG methods was extracted in the way it was, for the actual dataset. The prompts extracted were used here.

There were four methods implemented for tasks 2 and 3. For task 1, five models were implemented.

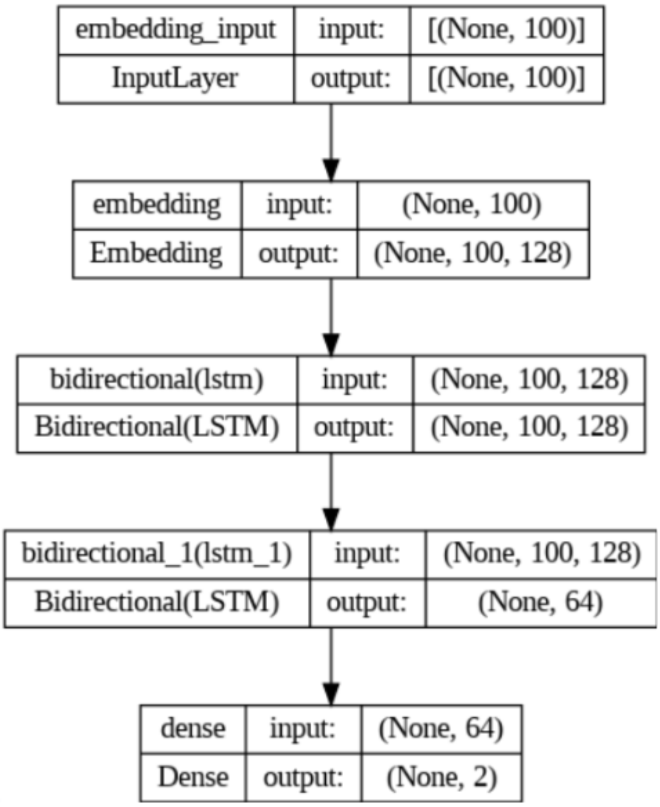
embedding_input | input: | [(None, 100)]
InputLayer | output: | [(None, 100)]
embedding | input: | (None, 100)
Embedding | output: | (None, 100, 128)
bidirectional(lstm) | input: | (None, 100, 128)
Bidirectional(LSTM) | output: | (None, 100, 128)
bidirectional_1(lstm_1) | input: | (None, 100, 128)
Bidirectional(LSTM) | output: | (None, 64)
dense | input: | (None, 64)
Dense | output: | (None, 2)

Figure 1: Method 1

3.2 Tokenizer + BiLSTM

The first model uses a normal tokenizer and BiLSTM model with the first layer consisting of 64 units and the next consisting of 32 units. In this method, the punctuations as well as the stopwords were removed at the preprocessing level. For the task 1 the two chat column's tokens were appended along the first axis.

3.3 BERT + BiLSTM

Coming to method 2 the previous method's BiLSTM architecture was utilized but this time the tokenizer used was the BERT tokenizer. In this case too the punctuations and the stopwords have been removed. The BERT tokens were padded to the maximum length. For the task 1 the two chat column's tokens were appended along the first axis.

3.4 Style + BiLSTM

In this model, the BiLSTM architecture used is the same as that of the 1 and 2 but the embedding dimension changes as the Stylometric features were used. The punctuations and the stopwords were not removed so as to preserve the syntactic style of the chat given. The BERT tokenizer has been used and the stylometric features were appended

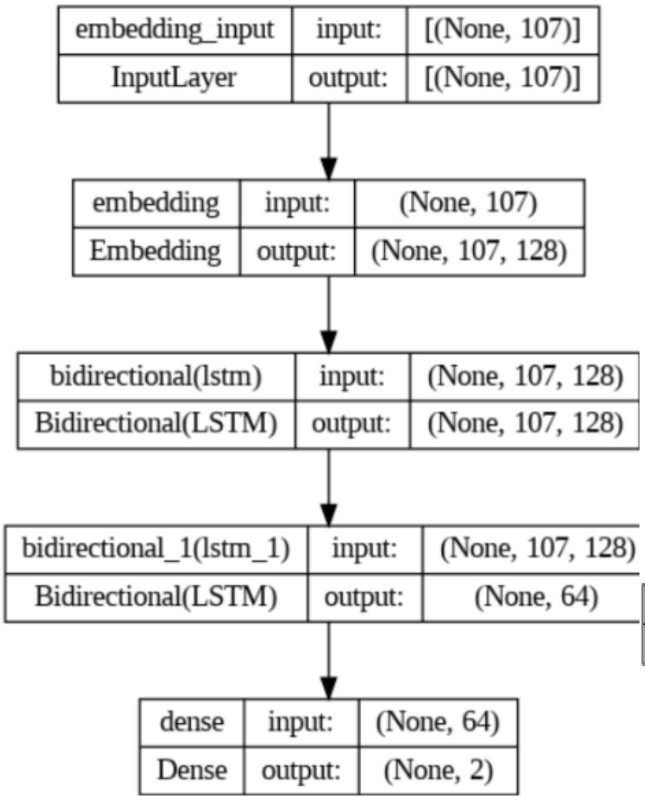
embedding_input | input: | [(None, 107)]
InputLayer | output: | [(None, 107)]
embedding | input: | (None, 107)
Embedding | output: | (None, 107, 128)
bidirectional(lstm) | input: | (None, 107, 128)
Bidirectional(LSTM) | output: | (None, 107, 128)
bidirectional_1(lstm_1) | input: | (None, 107, 128)
Bidirectional(LSTM) | output: | (None, 64)
dense | input: | (None, 64)
Dense | output: | (None, 2)

Figure 2: Method 3

to the BERT tokens after padding them to a maximum length. The stylometric features used were average word length, average sentence length, vocabulary size, lexical diversity, noun count, verb count, adjective count.

3.5 Style + BiLSTM + maxpooling

In this model, the stylometric analysis remains along with the BERT tokenizer but the BiLSTM architecture has been modified with the addition of global max-pooling, dropout and batch normalization layers. The BERT tokenizer was used in the similar fashion to that of the method 3 and stylometric features were appended to those tokens. The stylometric features used were average word length, average sentence length, vocabulary size, lexical diversity, noun count, verb count, adjective count. The BERT tokens were padded to maximum length before appending them with stylometric features. The global max-pooling layer is used to regularize the over-fitting where as the Batch normalization layer normalizes the output at each intermediate layer in order to keep it's probability distribution from getting distorted. The dropout layers ensure that the over-fitting is regularized.

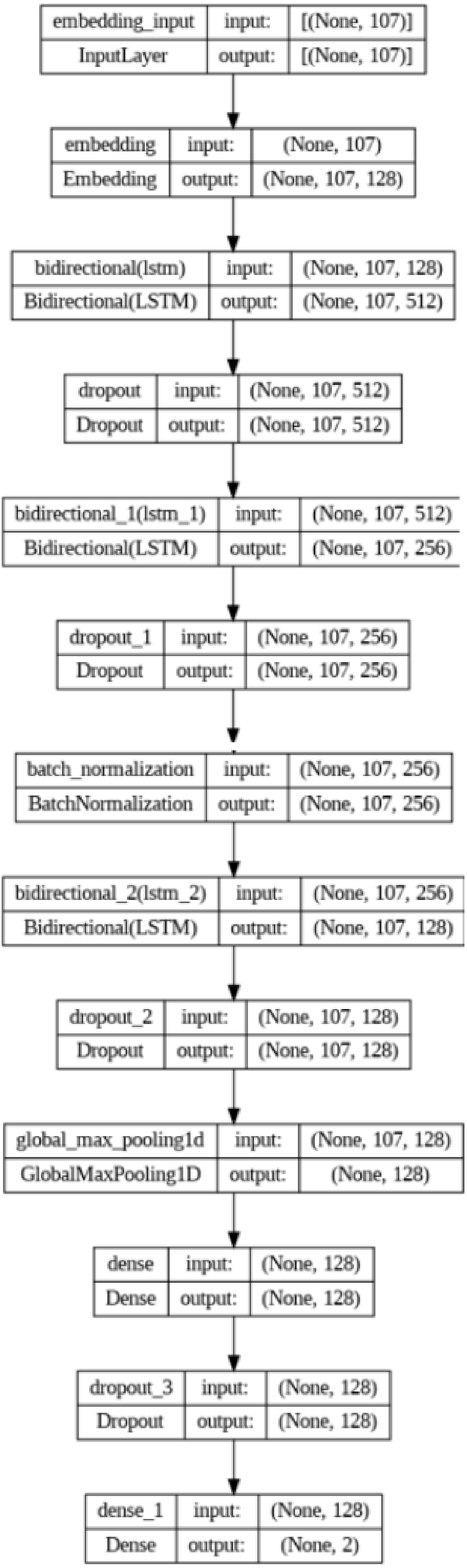
embedding_input | input: | [(None, 107)]
InputLayer | output: | [(None, 107)]
embedding | input: | (None, 107)
Embedding | output: | (None, 107, 128)
bidirectional(lstm) | input: | (None, 107, 128)
Bidirectional(LSTM) | output: | (None, 107, 512)
dropout | input: | (None, 107, 512)
Dropout | output: | (None, 107, 512)
bidirectional_1(lstm_1) | input: | (None, 107, 512)
Bidirectional(LSTM) | output: | (None, 107, 256)
dropout_1 | input: | (None, 107, 256)
Dropout | output: | (None, 107, 256)
batch_normalization | input: | (None, 107, 256)
BatchNormalization | output: | (None, 107, 256)
bidirectional_2(lstm_2) | input: | (None, 107, 256)
Bidirectional(LSTM) | output: | (None, 107, 128)
dropout_2 | input: | (None, 107, 128)
Dropout | output: | (None, 107, 128)
global_max_pooling1d | input: | (None, 107, 128)
GlobalMaxPooling1D | output: | (None, 128)
dense | input: | (None, 128)
Dense | output: | (None, 128)
dropout_3 | input: | (None, 128)
Dropout | output: | (None, 128)
dense_1 | input: | (None, 128)
Dense | output: | (None, 2)

Figure 3: Method 4

Model	**GPT-2**	**InstructGPT**	**Human**	**Xlnet**	**Fair**	**GPT**	**Ctrl**	**PPLM**	**Grover**	**GPT-3**	**XLM**	**Avg F1 Score**
Tokenizer + BiLSTM	0.94	0.40	0.96	0.54	0.45	0.33	0.48	0.57	0.74	0.97	0.93	0.66
BERT + BiLSTM	0.94	0.49	0.95	0.56	0.70	0.39	0.58	0.67	0.70	0.97	0.92	0.72
Style + BiLSTM	0.95	0.63	0.97	0.71	0.73	0.61	0.69	0.69	0.74	0.97	0.94	0.78
Style + BiLSTM + maxpooling	0.96	0.65	0.95	0.68	0.75	0.49	0.71	0.75	0.79	0.98	0.96	0.79
Random Forest	0.53	0.61	0.74	0.99	0.49	0.99	0.98	0.67	0.61	0.59	1.00	0.75
BERT+RF	0.14	0.63	0.45	0.40	0.18	0.61	0.20	1.00	0.17	0.52	0.67	0.45
Attention+RF	0.53	0.60	0.52	0.83	0.51	0.91	0.89	0.72	0.29	0.54	0.96	0.66

Table 1: Task 3 Results

3.6 GridsearchCV + XGBoost

The XGBoost with parameter grid which is a regularization parameter, gamma: This is the kernel coefficient for 'rbf', 'poly' and 'sigmoid', kernel: This is the kernel type to be used in the algorithm, learning rate, n estimators; is used along with the grid search CV which trains a XGBClassifier model for every combination of parameters in the grid and uses cross-validation to determine which combination results in the highest performance. The oversampling analysis is done using Synthetic Minority Over-sampling TEchnique (SMOTE).

4 Results

4.1 Task 1

On task 1, the five models have been trained upon both the balanced and the imbalanced sets. From the table it's clear that the GridsearchCV + XG Boost has given good results comparated with that of the other models. The results on the balanced dataset are quite good and that on the imbalanced set have exceeded those on the balanced set due to the SMOTE analysis.

Model	**Balanced (1:1)**			**Imbalanced (1:10)**		
	P	**R**	**F1**	**P**	**R**	**F1**
Tokenizer + BiLSTM	0.60	0.59	0.58	0.59	0.56	0.56
BERT + BiLSTM	0.55	0.55	0.54	0.58	0.55	0.55
Style + BiLSTM	0.48	0.48	0.44	0.36	0.50	0.42
Style + BiLSTM + maxpooling	0.56	0.53	0.44	0.84	0.56	0.53
GridsearchCV+ XG Boost	0.84	0.79	0.81	0.92	0.86	0.89

Table 2: Task 1 Results

4.2 Task 2

For task 2, four models were implemented out of them the style + BiLSTM + maxpooling model has performed well on both the balanced and the imbalanced datasets.

Model	**Balanced (1:1)**			**Imbalanced (1:10)**		
	P	**R**	**F1**	**P**	**R**	**F1**
Tokenizer + BiLSTM	0.82	0.82	0.82	0.80	0.64	0.68
BERT + BiLSTM	0.79	0.75	0.77	0.73	0.81	0.76
Style + BiLSTM	0.85	0.84	0.84	0.86	0.85	0.85
Style + BiLSTM + maxpooling	0.85	0.85	0.85	0.87	0.83	0.85

Table 3: Task 2 Results

4.3 Task 3

For the task 3, four models were implemented and compared with the three Random Forests models which were a simple Random Forests model, one with BERT tokenizer and another one with the attention mechanism, among which the style + BiLSTM + maxpooling model has performed well in the case of GPT2, InstructGPT, FAIR, GROVER, GPT3 and also the overall F1 macro score. The style + BiLSTM model has performed well on the human generated chats. Where as on the rest either the Random Forests model or the BERT + random forests models have performed well.

4.4 Comparison with milestone 2

The performance of the models was consistent. On task 1 the overall F1 score achieved was 0.40 on the imbalanced set but the F1 score achieved on the same during milestone 3 was 0.89. This might be due to the SMOTE analysis with XGBoost method. Coming to the task 2 the scores range was 0.06 to 0.09 which has been greatly improved to 0.85 which might be due to the stylometric features and BERT tokenizer. On task 3 the scores were in the range of 0.04 to 0.08, which has got increased to 0. The improved ranges were from 0.40 to 0.98. This might be due to the fine tuning 0f the BiLSTM architecture along with maxpooling to contol the overfitting and stylometric features with BERT tokenizer.

4.5 Reddit Task 1

Coming to the Reddit case study, for task 1 the XG Boost model has given good results on the balanced as well as the imbalanced datasets. For the task 2, BERT + BiLSTM has given fair results which might be the case of overfitting but the style + BiLSTM model has also performed quite well.

Model	Balanced (1:1)			Imbalanced (1:2)		
	P	R	F1	P	R	F1
Tokenizer + BiLSTM	0.56	0.56	0.56	0.58	0.54	0.51
BERT + BiLSTM	0.62	0.62	0.61	0.64	0.64	0.54
Style + BiLSTM	0.68	0.66	0.65	0.69	0.64	0.65
Style + BiLSTM + maxpooling	0.70	0.65	0.63	0.85	0.55	0.51
GridsearchCV+ XG Boost	0.77	0.75	0.76	0.92	0.86	0.89

Table 4: Reddit Task 1 results

4.6 Reddit Task 2

For the task 2, BERT + BiLSTM has given fair results which might be the case of overfitting but the style + BiLSTM model has also performed quite well.

Model	Balanced (1:1)			Imbalanced (1:2)		
	P	R	F1	P	R	F1
Tokenizer + BiLSTM	0.95	0.91	0.92	0.91	0.94	0.92
BERT + BiLSTM	1.00	1.00	1.00	1.00	0.99	0.99
Style + BiLSTM	0.97	0.97	0.97	0.98	0.96	0.97
Style + BiLSTM + maxpooling	0.94	0.94	0.94	0.96	0.92	0.94

Table 5: Reddit Task 2 results

4.7 Reddit Task 3

Coming to the task 3 of reddit case study, the style + BiLSTM + maxpooling layer has performed well on the GPT3 and human generated chats and on the InstructGPT it has performed fairly compared to that of the BERT + BiLSTM which has performed well. On the whole the style + BiLSTM + maxpooling model has performed well.

Model	GPT-3	Instruct GPT	Human	Average
Tokenizer + BiLSTM	0.69	0.86	0.63	0.73
BERT + BiLSTM	0.73	0.97	0.65	0.78
Style + BiLSTM	0.71	0.95	0.63	0.77
Style + BiLSTM + maxpooling	0.75	0.93	0.68	0.79

Table 6: Reddit Task 3 results

5 Discussion and Error Analysis

5.1 Task 1

The tokenizer + BiLSTM model performs better on balanced dataset compared to that of the imbalanced dataset showing that it might be prone to class imbalance. The BERT + BiLSTM model has shown that it performs better on the class imbalance compared to that of the first model. The style + BiLSTM model doesn't perform well compared the previous two. The style + BiLSTM + maxpooling model performs well on the imbalanced dataset. But as the recall is less on the balanced set is more compared to that of the imbalanced set, it might be overfitting on the imbalanced set with respect to one class. The XGboost method performs well on both the sets. The model is robust to the class imbalance and generalizes well.

The first model has lot of area left for the improvement. The class imbalance must be handled with the techniques using the undersampling or oversampling or SMOTE analysis. Where as for the second method the fine tuning on the BERT could have improved the results. In the third method, there is scope for improvement in handling the class imbalance and features on style. In the fourth model, the hyper-parameter tuning could be further optimized. The same is for the fifth method. On the whole the methods on task 1 could be improvised over the class imbalance with techniques such as the SMOTE, oversampling and undersampling.

5.2 Task 2

On the balanced set, all the models have performed well with the F1 scores from 0.77 to 0.85 and the style + BiLSTM + maxpooling method has achieved the highest score. But on the imbalanced set the the performance differs significantly. The style + BiLSTM + maxpooling model has still achieved the highest F1 score but recall has dropped to 0.83. The BERT + BiLSTM model has performed well on the imbalanced set with significance.

The tokenizer + BiLSTM has a lot of scope for improvement. This might be due to the lack of sufficient training on the minority class and the need for better hyperparameter tuning.

5.3 Task 3

Here the highest F1-score is achieved by the Style + BiLSTM + max-pooling model with a score of 0.79. Random Forest has a high F1-score of 1.00, indicating that it is overfitting to the data on XLM. The results suggest that models of Random Forests and the , as seen in the higher F1-scores of RF and Style + BiLSTM + max-pooling.

From the table it is clear that the style + BiLSTM + maxpooling model has performed well with respect to the overall F1 score. It has also performed well on the InstructGPT, GPT2, FAIR, GROVER, GPT3 and also gave near to good results over the data generated by human. Not only that almost all the models have given near to good results on the chats generated by the humans and even on the GPT2, GPT3, XLM. On InstructGPT the results have improved consistently from the first to the fourth model. This is the same for the PPLM, CTRL. The GROVER is said to be one of the toughest datasets[1] on which the style + BiLSTM + maxpooling model was able to perform quite good.

Improvement could be done on models especially, fine tuning of the stylometric features and more complex style analysis could be done. Along with these syntactic analysis, the semantic analysis also would be helpful in classifying the generated chat. More analysis could be done with fine tuning over the model architecture of the BiLSTM with the max-pooling layers.

5.4 Reddit Task 1

The four models have varying levels of performance on balanced and imbalanced datasets, with different strengths and weaknesses. The Tokenizer + BiLSTM model struggles with identifying the minority class in imbalanced datasets and could benefit from oversampling or undersampling techniques. The BERT + BiLSTM model performs reasonably well but could improve by further fine-tuning BERT or exploring different combinations with BiLSTM. The Style + BiLSTM model shows good performance on both datasets, but there's room for improvement by tuning the BiLSTM architecture and exploring different style features. The Style + BiLSTM + maxpooling model has high precision but low recall and F1-score, potentially due to maxpooling, and could benefit from adjusting the strategy or integrating style features differently. All models could benefit from hyperparameter tuning, feature engineering, and ensemble methods.

5.5 Reddit Task 2

The four models perform well on both the balanced and imbalanced datasets. The Tokenizer + BiLSTM model has good performance, but optimizing the tokenization process or fine-tuning the BiLSTM's hyperparameters could improve it. The BERT + BiLSTM model performs better than all the other models and achieved near to good scores on all metrics, but fine tuning BERT could lead to even better performance. The Style + BiLSTM model performs very well, but class imbalance may be impacting its precision and recall, and techniques to handle class imbalance could be explored. The Style + BiLSTM + maxpooling model performs similarly to the third, and incorporating different maxpooling strategies or other ways of handling class imbalance could be explored to further improve it. Overall, the combination of the specific features with BiLSTM is effective, but there is always scope for fine tuning and experimentation.

5.6 Reddit Task 3

The four models evaluated show promising results, but there is still scope for improvement in handling GPT-3 and human text. The Tokenizer + BiLSTM model performs relatively poorly compared to the other models and could benefit from enhanced tokenization techniques. The BERT + BiLSTM model performs better but still has room for improvement, possibly through more domain-specific training or different ways to combine BERT embeddings with BiLSTM. The Style + BiLSTM model performs similarly to BERT + BiLSTM but may benefit from experimenting with different style features. The Style + BiLSTM + maxpooling model shows the best average performance, but further experimentation with sequence aggregation methods may improve performance on GPT-3 and human text. In general, exploring feature engineering, model architectures, and fine-tuning strategies could lead to further improvement.

6 Conclusion

The models implemented have performed well in some particular areas. In particular the models with the stylometric features have performed well in tasks 2 and 3 but not that well in case of task 1 when compared to that of the XG Boost model. Again comparing to the Random Forests models in task 3 the max-pooling layer has proven to be effective over the Random Forests in case of some models and also it has proven well on regularization.

The BERT tokenizer has proven to be extremely useful on the as the models used along with them have given good results. The style analysis combined with the POS estimations such as the noun count, verb count and the adjective count has given good insight into the syntactic structure of the chat generated by each NLG method or the human.

7 Contribution

Myself and one of my teammates have collected the data from the GitHub repository and using the prompts from the GitHub repository we have generated the texts from the InstructGPT and GPT3 using the OpenAI key. Also I, alone have worked on preparing the datasets for the tasks 1 and 2 which I have prepared two sets for each task, one for balanced and the other for imbalanced cases. I have also implemented the the Style + BiLSTM + maxpooling model on all the three tasks. Also implemented the Random Forests model with BERT tokenizer and Random Forests with Attention mechanism in task 3 for the comparison purpose along with one of my teammates. I have also implemented the Style + BiLSTM + maxpooling model over the three tasks of the Reddit case study. In total I have implemented 11 code files for all the tasks both the actual and the Reddit study combined in addition to preparing the datasets for task 1 and 2.

References

[1] Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. *Authorship Attribution for Neural Text Generation*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8384–8395, Online. Association for Computational Linguistics, 2020.

[2] Dainis Boumber, Yifan Zhang, and Arjun Mukherjee. *Experiments with Convolutional Neural Networks for Multi-label Authorship Attribution*. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), 2018.

[3] F. Jafariakinabad and K. A. Hua. *Style-Aware Neural Model with Application in Authorship Attribution*. In 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 2019, pp. 325–328. doi: 10.1109/ICMLA.2019.00061.

[4] Haiyan Wu, Zhiqiang Zhang, and Qingfeng Wu. *Exploring Syntactic and Semantic Features for Authorship Attribution*. Applied Soft Computing, Volume 111, 2021, 107815, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2021.107815.

[5] Stamatatos E. *A Survey of Modern Authorship Attribution Methods*. J. Am. Soc. Inf. Sci. Technol. 60(3) (2009) 538–556.

[6] R. Zhang, Z. Hu, H. Guo, Y. Mao. *Syntax Encoding with Application in Authorship Attribution*. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 2742–2753.

[7] Jafariakinabad F., Tarnpradab S., Hua K.A. *Syntactic Recurrent Neural Network for Authorship Attribution*. 2019, arXiv preprint arXiv:1902.09723.

[8] Q. Li, Z. Li, J.-M. Wei, Y. Gu, A. Jatowt, Z. Yang. *A Multi-Attention Based Neural Network with External Knowledge for Story Ending Predicting Tasks*. In Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1754–1762.

[9] Ruder S., Ghaffari P., Breslin J.G. *Character-level and Multi-channel Convolutional Neural Networks for Large-scale Authorship Attribution*. CoRR abs/1609.06686 (2016).

[10] Elisa Ferracane, Su Wang, and Raymond Mooney. *Leveraging Discourse Information Effectively for Authorship Attribution*. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 584–593, 2017.

[11] R. Caruana and A. Niculescu-Mizil. *An Empirical Comparison of Supervised Learning Algorithms*. In Proceedings of the 23rd International Conference on Machine Learning, pp. 161–168, 2006.

[12] M. Khonji, Y. Iraqi and A. Jones. *An Evaluation of Authorship Attribution Using Random Forests*. In 2015 International Conference on Information and Communication Technology Research (ICTRC), Abu Dhabi, United Arab Emirates, 2015, pp. 68–71. doi: 10.1109/ICTRC.2015.7156423.

[13] J. Gaston et al. *Authorship Attribution vs. Adversarial Authorship from a LIWC and Sentiment Analysis Perspective*. In 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 2018, pp. 920–927. doi: 10.1109/SSCI.2018.8628769.

[14] Yunita Sari, Mark Stevenson, and Andreas Vlachos. *Topic or Style? Exploring the Most Useful Features for Authorship Attribution*. In Proceedings of the 27th International Conference on Computational Linguistics, pages 343–353, Santa Fe, New Mexico, USA. Association for Computational Linguistics, 2018.

[15] Hugo Jair Escalante, Thamar Solorio, and Manuel Montes-y Gomez. *Local Histograms of Character N-grams for Authorship Attribution*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, pages 288–298, Stroudsburg, PA, USA. Association for Computational Linguistics, 2011.

[16] Ahmed Abbasi and Hsinchun Chen. *Writeprints: A Stylometric Approach to Identity-level Identification and Similarity Detection in Cyberspace*. ACM Transactions on Information Systems (TOIS), 26(2), 2008.

[17] Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. *Authorship Attribution with Topic Models*. Journal Computational Linguistics, 40(2), 2013, pp. 269–310.