Team: JCL.

1. Introduction

The advancements in NLP have enabled machines to produce human-like text. This, while groundbreaking, has raised concerns related to content authenticity, accuracy, and authoritativeness. The objective of this project is to develop a machine learning model distinguishing between human and machine-generated texts. Two challenges were presented in the task. Firstly, the data is sourced from two domains: Domain 1 and Domain 2. Each domain has its distinct characteristics, such as human-generated texts tend to be shorter than machine-generated texts in Domain 1, with the average length of machine and human-generated texts being 52 and 27, respectively. In contrast, the reverse is observed in Domain 2, 146 for machine and 216 for human-generated text length. Secondly, class labels are imbalanced in Domain 2. There are 2150 human-generated texts, significantly lower than 12750 machine-generated texts. Typically, we would ideally have balanced representations from each domain for training. However, the imbalance of Domain 2 makes the classification challenging. The model training in Domain 2 may be biased towards certain patterns, which may be the majority, leading to the model underperforming on human text from Domain 2 during testing, especially if the test set is more balanced. In this report, we aim to address the two issues by leveraging NLP approaches with different NLP word representations accompanied by different strategies for the AI-text detection task.

2. Method

In our final approach, we adopted the Bidirectional LSTM (BiLSTM) as the primary model for the classification task. Given the challenges posed by class imbalance in the dual-domain dataset, we employ a hybrid approach that combines the Domain-Adversarial Training of Neural Networks (DANN) with a class re-weighting strategy.

2.1 BiLSTM

Our goal is to detect whether a text input is generated by a human or a machine, which can be seen as a binary text classification task. In this case, the text meaning and the order of the tokens in a text are critical. Common techniques such as recurrent neural networks (RNN) can handle this kind of text classification well, as they inherently process sequences token by token, preserving the data's order and context. Among them, the BiLSTM has emerged as one of the most effective approaches for such tasks. The BiLSTM model operates by processing the input data in both forward and backward directions, capturing information from both past and future states. Given that we are discerning between AI-generated and human-generated text, understanding the context from both the preceding and succeeding tokens in the sequence aids in capturing subtle nuances and patterns that can differentiate between the two.

2.2 Hybrid Strategy: Class Re-weighting + DANN

Considering the test dataset consisting of Domain 1 and Domain 2 with balanced class distribution, we adopt the Data Augmentation (See section 3 for more details.) by combining Domain 1 and Domain 2 as the training dataset in order to gain better performance with more data samples. However, Domain 2 has imbalanced class distributions, which will cause issues with the text classification performance. Thus, we first focused on the class imbalance issue, and after testing the different strategies (See section 3 for more details.), we adopted the Class Re-weighting into our final approach. After choosing the strategy for handling the class imbalance issues, considering our training dataset comes from two different domains, we planned to add the domain adaptation approach to make our classification model more robust. We finally adopted the DANN into our final approach, and the performance shows the best among our candidate strategy combinations (See section 3 for more details).

2.2.1 Class Re-weighting

In the Domain 2 dataset, class label 0 has significantly more data points than the class label 1, which apparently exists the class imbalance issue and label 1 is as the minority class. If we directly apply our BiLSTM model could result in a classifier skewed towards the majority class. To solve the imbalance dataset problem, we employ a class re-weighting strategy. Specifically, the weights for each class are computed based on their frequency in the training set, and these weights are incorporated into our Binary Cross-Entropy with Logits Loss function in our BiLSTM model. The loss function in our model can be described below:

$$l = -w\,[\,p \cdot y \cdot log\sigma(x) + (1-y) \cdot log(1-\sigma(x))]$$

By increasing the weights p for minority class, the model receives a larger loss for class label 1, allowing the model to learn more from the minority class.

2.2.2 Domain-Adversarial Training of Neural Networks (DANN)

Common strategies for training data from different domains include re-sampling techniques, synthetic data generation, and domain adaptation. Among these methods, the domain adaptation has demonstrated efficacy[1] in

classification tasks. In the real-world scenarios such as fake-news detection from multi-source domains always utilize the powerful NLP deep learning methods, and combining DANN method, which is well-fit for deep neural network architectures. Given we adopt the BiLSTM model for our classification task, we apply the DANN technique as for the domain adaptation experiment.

The main idea of DANN is to ensure that the neural network model learns features that are agnostic to the domain, thus improving generalization across different domains. This is achieved by incorporating an adversarial component into the training. Based on this, our DANN approach includes the following three parts:

- **Feature Extractor**: It is a neural network component that is trained to transform input data into a representation that's useful for the main task (human vs. machine-generated text in our case).
- **Label Predictor**: This predicts the actual label of the data using the features provided by the feature extractor. This is the output layer of our BiLSTM to classify the text as either human or machine-generated.
- **Domain Classifier(Discriminator)**: The adversarial part of DANN. Its task is to predict the domain of the input data based on the representations returned from the feature extractor.

During training, the goal of the feature extractor is twofold: to help the label predictor with its task while "fooling" the domain classifier. We add a gradient reversal layer to achieve this. Before the gradients from the domain classifier are back-propagated to the feature extractor, they are multiplied by a negative value (effectively reversing them). This ensures that the feature extractor is discouraged from producing domain-specific features. Instead, it gets trained to generate features that are domain-agnostic.

3. Experimental Result and Discussion

Our experiment consists of two parts: the baseline approach experiment and the BiLSTM approach experiment. The baseline approach is based on the bag-of-words (BoW) text representation as model input, while the BiLSTM approach is based on the word embedding techniques, which need more training resources to acquire and optimize the dense parameters. We adopted the Logistic regression and SVM as the baseline classification models since they are easy-implemented and perform adequately for text classification tasks. We first started with these baseline approaches and combined them with different strategies in order to get quick results for further improvements. These strategies include **weighted average ensemble** and **data augmentation** for dealing with dual-domain and class imbalance challenges. Then, we pick up the feasible strategies for further application to the BiLSTM approach experiment. For each experiment, we split 80% of the defined training dataset for training and the remaining 20% for testing. The result summary is listed as follows:

Category	Approach	Accuracy	F1 score	Kaggle Accuracy
Baseline	Logistic Regression Domain2	0.470	0.355	-
	Logistic Regression weighted average ensemble	0.717	0.701	-
	Logistic Regression data augmentation	0.840	0.826	0.738
	SVM+ weighted average ensemble	0.652	0.395	-
	SVM+ data augmentation	0.707	0.670	0.604
BiLSTM	BiLSTM+Data Augmentation+oversampling	0.901	0.798	0.754
	BiLSTM+Data Augmentation+Re-weight	0.8392	0.8328	0.818
	BiLSTM+DANN+Re-weight (final approach)	0.8612	0.8519	0.83

Table 1: Experiment Result

3.1 Baseline Approach

For the Logistic Regression experiment, since Domain 2 captures more features than Domain 1, we started with the simplest singular model trained on Domain 2 and then tested on Domain 1. However, the model yields an accuracy below 0.5. This disparity was due to a dual-domain problem, as the features identified as significant in Domain 2 may not hold relevance in Domain 1. To address this challenge, we explored two strategies. The first employs a **weighted average ensemble**, wherein individual models trained on each domain have their predictions aggregated, with weights assigned based on their respective accuracies. The second strategy, **data augmentation**, combines the data from both domains to train a singular model. Experimentation reveals superior performance from the latter method, achieving a Kaggle accuracy of 0.738 (See Table 1.). The reason can be that holistic training data introduces a broader spectrum of patterns, ensuring consistent feature recognition and reducing overfitting tendencies. Another BoW approach involves Support Vector Machines (SVM). Although SVM can capture non-linear relationships in the data, it is time-intensive for training and parameter tuning. To combat this, we truncated the text features, retaining only the initial 200 features for both datasets. However, its permanence is still poorer compared to the original BoW approach.

Despite the baseline's simplicity and comprehensibility, the BoW model overlooks word sequences, missing out on context and semantics, leading to unstable performance. Thus, we experimented with the deep learning approach with word embedding techniques.

3.2 BiLSTM Approach

From the baseline approach results, we observed that the **Data Augmentation** approach is beneficial to the classification task as our target dataset consists of two domains. Hence, we combine both of the domains as the training dataset. However, as our number of data samples increased, the data imbalance issue still exists. This, we firstly focused on the issue of imbalanced class distribution in Domain 2. We consider the oversampling method and the class re-weight strategies in our experiment to deal with the issue.

3.2.1 The Oversampling Strategy Result

We adopted the random oversampling method that is added to the BiLSTM model. In the training process, a weighted random sampler is used to select data points randomly while ensuring that the class distribution remains consistent between the two classes. Table 1 shows that using the oversampling strategy produces both a high accuracy of 0.90 and an F1 score of 0.798 by balancing the dataset. However, the Kaggle accuracy remains relatively low, with a score of 0.754. This may be due to overfitting caused by the random oversampling method over the same dataset, which duplicates the minority class data points.

3.2.2 The Class Re-weighting Strategy Result

We attempted another strategy to address the challenge of data imbalance while avoiding overfitting by using the class re-weighting approach. By altering the weight p for classes for the loss function calculation, the model can averagely learn from the minority class and avoid concentrating on learning from the majority class, thus avoiding the overfitting and bias issue. We discovered that this method yielded a greater F1 score with 0.8328 compared to oversampling and had a higher accuracy of 0.818 on the Kaggle test dataset. Therefore, we decided to use the class re-weighting strategy as the final strategy for experiment to address the class imbalance issue.

3.2.3 Final Approach Result

Finally, we tested our hybrid strategy: the class re-weighting and DANN into our BiLSTM classification models. Our model was trained for 23 epochs with an Adam optimizer. The gradient reversal scale (alpha) was set at 7, and the gradients were clipped at a value of 1 to prevent exploding gradients. Throughout the training process (See Figure 1), both the classifier and domain discriminator improved in terms of loss, accuracy, and F1-score, indicative of a balanced performance across classes. The domain discriminator, aimed at reducing domain-specific biases, showed an accuracy trend around 0.54 to 0.58. It implies that our model has learned to create somewhat domain-invariant features, which is the objective when aiming for domain-invariant features. In summary, our final approach adequately addressed the issue of class imbalance and showed promising results in making the feature representations more domain-agnostic. The accuracy of 0.83 score on the Kaggle test set indicates the robustness and generalizability of our model. However, the classifier still has room to improve, as Figure 1 shows the accuracy and F1 score comes to converge. More sophisticated methods can be considered such as setting up the dynamic learning rate or dynamic gradient reversal scale mechanism.

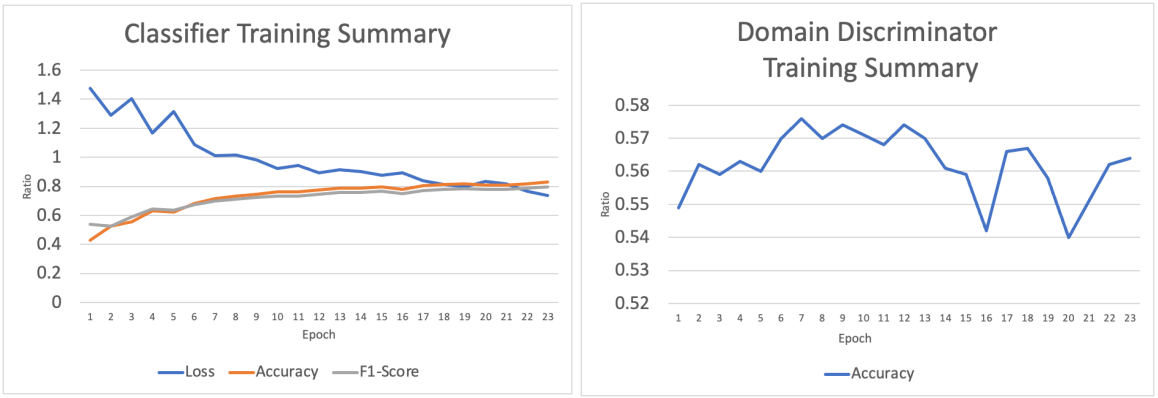

Figure 1: The Training Summary

4. Conclusion

In this project, we adequately developed a BiLSTM-based machine learning model to differentiate between human and machine-generated text. We faced two main challenges: the distinctive characteristics of two domains and the class imbalance in Domain 2. By adopting a hybrid strategy that combined DANN with a class re-weighting approach, we were able to address both issues effectively with adequate performance. However, More sophisticated methods can be considered for the BiLSTM model and DANN, since this combined approach will be a trade-off between classifier and domain discriminator performance.

Reference

[1] Yuan, Hua, Jie Zheng, Qiongwei Ye, Yu Qian, and Yan Zhang. "Improving fake news detection with domain-adversarial and graph-attention neural network." *Decision Support Systems* 151 (2021): 113633.