# Enhancing Text Classification in Imbalanced Domains: A Multimodal Approach

## Setup informations

In this project, we partitioned the dataset into an 80% portion for training and a 20% portion for validation across all model training processes. Test accuracy refers to the outcome of predictions submitted to Kaggle. The primary metrics for assessing our model's performance encompass the F1 score, validation accuracy, and test accuracy. The data's inherent imbalance presented a challenge in building a single model for this project. To address this issue, we employed multiple models with varying complexity.

## Data inspection

We observed variations in text length between human-generated and machine-generated data in both domains. This observation suggests that text length could serve as a crucial feature for model training (Fig.1 A). In Domain 2, we encountered data imbalance, where the amount of human-generated data was significantly lower than that of machine-generated data. Furthermore, the machine-generated data from different models exhibited uneven distribution (Fig.1 B).

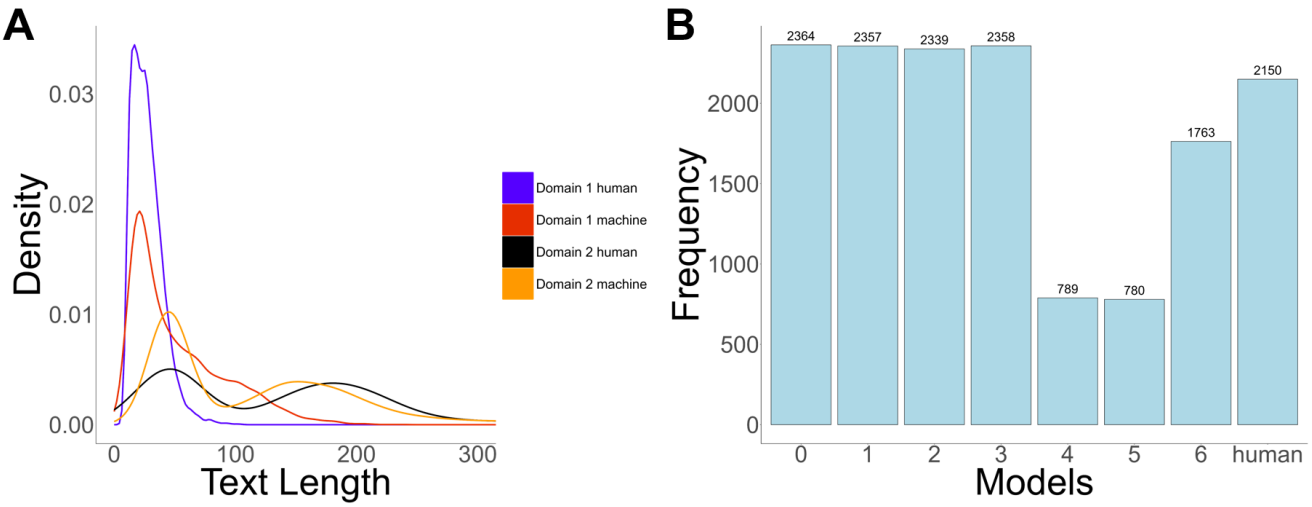


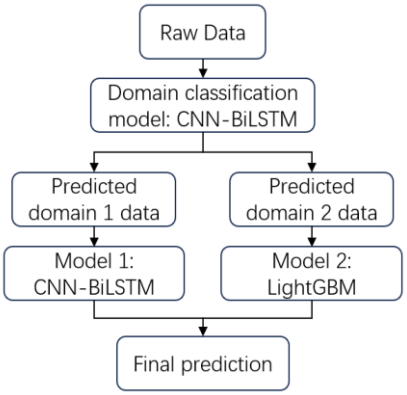

Figure 1. Data distribution. (A) the distribution of machine and human sentence length in both domains. (B) Histogram of data number from different models in domain 2.

## Hypothesis setup

Our initial hypothesis assumed that both machine-generated and human-generated sentences have distinct domain-independent characteristics. Additionally, we considered the possibility of augmenting Domain 2's human-generated data by integrating it with Domain 1 due to their assumed similarity. To explore this, we combined Domain 1 and Domain 2, creating our training dataset for the baseline model. Using logistic regression (LR) with TF-IDF vectorization, the model achieved an impressive 86% accuracy on the validation dataset. However, its test performance drastically declined to 68%.

Further evaluation focused on LR's performance in Domain 1 and Domain 2. When classifying machine-generated sentences, LR demonstrated high accuracy in both domains (92.36% in Domain 1 and 84.40% in Domain 2). Notably, there was a substantial accuracy difference between the two domains. Calculating the F1 score revealed an unusual performance pattern in Domain 2 (F1 score 0.03 for human). Consequently, we concluded that human-generated data in Domain 2 differs significantly from that in Domain 1, as evident in Figure 1A, where Domain 1 predominantly contains shorter sentences compared to Domain 2. Ultimately, the augmentation-by-integration approach proved ineffective.

Given the inconsistent model performance on both machine-generated data in both domains and the poor performance in classifying human data in Domain 2, we deemed our initial hypothesis inappropriate. Consequently, we established three goals to enhance predictive performance (models see Fig. 2):




**Aim 1.** Data from Domain 1 and Domain 2 should be classified separately, necessitating accurate domain classification.

**Aim 2.** For Domain 1, where data is ample and balanced, complex models should be considered.

**Aim 3.** Models robust to imbalanced data should be trained for Domain 2.

Figure 2. Prediction program setup

**Aim 1. Domain classification models**

For the domain classification model (DCM), we concatenated data from Domain 1 and Domain 2 for training. Initially, we employed the LR model as our DCM baseline, achieving commendable performance at 96.96%. To enhance predictive accuracy, we trained a Bidirectional Long Short-Term Memory (Bi-LSTM) model, renowned for its excellence in text classification tasks[1]. The BiLSTM processes input data in both forward and backward directions, generating hidden states from both, thereby enhancing its contextual understanding. However, the tradeoff is that the BiLSTM model is computationally intensive. Taking inspiration from existing research and leveraging the dimensionality reduction capabilities of Convolutional Neural Networks (CNN), we developed a Convolutional Neural Network-Bidirectional Long Short-Term Memory (CNN-BiLSTM) model to serve as our DCM (Fig. 3)[2].

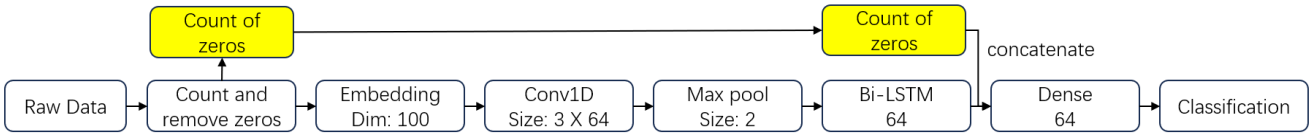



Figure 3. CNN-BiLSTM model structure

For the input text data, we removed zero values as they correspond to unknown words. To retain the information associated with these unknown words, we preserved the zero counts as a feature. We introduced a convolutional layer with a window size of 3, akin to capturing trigrams, to reduce data dimensionality while retaining critical information. The downsized output was then connected to the Bi-LSTM. Subsequently, we concatenated the zero counts (highlighted in yellow in Fig 3) with the Bi-LSTM output and passed them through a dense layer. We employed binary cross-entropy loss for our binary classification task. To mitigate overfitting, a dropout rate of 0.3 was applied to each layer in the model. The CNN-BiLSTM model surpassed the LR baseline, achieving an accuracy of 98.9%.

**Aim 2. Optimization of domain 1 classification using complex model (Model1)**

As part of our optimization efforts for domain 1 classification, we first established a baseline using a LR model, which yielded an accuracy of 92.36%. To further enhance the model's complexity, we then trained a CNN-BiLSTM model on the domain 1 data. Both models exhibited impressive accuracy, with the CNN-

[1] "Bidirectional LSTM with attention mechanism and convolutional ...." 14 Apr. 2019, https://www.sciencedirect.com/science/article/abs/pii/S0925231219301067. Accessed 21 Sep. 2023.
[2] (2015, November 27). [1511.08630] A C-LSTM Neural Network for Text Classification - arXiv. Retrieved September 21, 2023, from https://arxiv.org/abs/1511.08630

BiLSTM model outperforming the LR model, achieving an accuracy of 95.36%. This outcome aligns with our objective of demonstrating that increased model complexity, particularly when applied to datasets that are both sufficient and balanced, leads to improved accuracy.

**Aim3. Exploring models for domain 2 classification (Model2)**

**1. Change of loss function**

Our initial attempt involved applying the CNN-BiLSTM model to the domain 2 data, resulting in high accuracy but accompanied by a remarkably low F1-score (0.11). This was attributed to the challenges posed by the imbalanced dataset. In an effort to address this issue, we introduced weighted cross-entropy as the loss function; however, the outcomes were still unsatisfactory.

**2. Up-sampling and Down-sampling**

In our pursuit of a solution, we experimented with up-sampling the domain 2 data by randomly replicating the human-generated texts from domain 2 for training. Unfortunately, this approach led to overfitting, characterized by high training accuracy (81.41%) but low validation accuracy (75.52%) and a low F1-score (0.12). Additionally, we explored down-sampling the data by randomly removing text from the AI-generated texts to create a balanced training set. Regrettably, this strategy also yielded unfavorable outcomes (F1: 0.15), possibly due to the scarcity of machine-generated data, resulting in underfitting.

**3. Multitask learning**

Given the existence of model labels ranging from 0 to 6 in the domain 2 data, we tried to extract additional information by employing multi-task learning. We introduced an additional category classification output to predict the model label, with the intention of leveraging the interplay between the two tasks to enhance performance. Regrettably, this approach did not yield satisfactory results.

**4. Lightgbm**

After several trials, we postulated that the challenges lay in the complexity of NN models and the need for higher-quality data. Consequently, we explored less complex models, starting with SVM using TF-IDF vectorization. Remarkably, SVM outperformed the NN model. Inspired by the paper[3] using a boosting method to tackle the imbalanced dataset, we turned to LightGBM, a tree-based learning algorithm akin to random forests but characterized by enhanced speed and accuracy. LightGBM builds trees sequentially, with each tree correcting the errors of its predecessor. Notably, it adopts leaf-wise growth, emphasizing the expansion of individual leaves rather than level-wise growth. To find the best hyper parameter, we employed Bayesian Optimization, a faster alternative to grid search, which identifies optimal hyperparameters through probabilistic modeling rather than exhaustive search. A key parameter we tuned was the ‘class weight’ to address the dataset's imbalance. We utilized Bag-of-Words (BOW) as input for training, resulting in a 0.42 F1-score on five-fold cross-validation. It is important to note that BOW lacks the capability to capture sequential information within the data, leading to the loss of data insights. In response, we experimented with the 2-grams method to capture sequential information. However, this approach generated a word count matrix with 260,000 columns, surpassing the computational resources at our disposal. Situation was the same when we tried principal component analysis for dimensionality reduction on this extensive dataset. Therefore, it is theoretically possible to further optimize our model through the n-grams method.

**Summary of Our Prediction Program**

Recognizing the limitations of a single model in achieving accurate predictions, we devised a multi-component approach (Fig. 2). First, we trained a CNN-BiLSTM model to classify sentence domains (Acc: 98.8%). Second, we employed another CNN-BiLSTM model to classify sentences within Domain 1 (Acc: 95.36%). Lastly, for sentences in Domain 2, we trained a LightGBM model (F1: 42.7, Acc: 81%). Our final prediction is a composite outcome, combining the predictions from components 2 and 3 (Acc: 82.6%).

[3] (2004, June 1). Learning from imbalanced data sets with boosting and data generation. Retrieved September 21, 2023, from https://dl.acm.org/doi/10.1145/1007730.1007736