Two Systems, One Model: A framework for two-domain AI text detection

2024 S1 COMP90051 Assignment 1: Group 51

Lang (Ron) Chen *(1181506)* Un Leng Kam *(1178863)* Di Wu *(1208784)*

1 Introduction

The recent rise of AI text generation services brought urgency to developing human-AI text detectors to preserve integrity; cross-domain learning is also important as abundance of labelled data for specific domains are never guaranteed. This report investigates a Machine Learning (ML) approach for this task with text data from 2 domains.

The experiment dataset originates from two different domains; Domain 1 is balanced with 5000 samples in total, while domain 2 contains 1500 human and 11500 AI-generated samples. All text were tokenized and mapped into indices from 1 to 83582. Given the imbalanced nature of the dataset (overall 77%, AI 22% human), a stratified 70%-15%-15% train-val-test split was performed on the data based on both class and domain.

Our experiments identified the **dual imbalance in global and domain-specific labels** and **cross-domain inflicted noise** to be the main factors negatively impacting ML model performance on this task, and propose a training mechanism involving **domain-specific losses** (hence two systems) and **domain adversarial learning** to improve performance.

2 System Description

2.1 Transformer and BERT

The architecture of Bidirectional Encoder Representations from Transformers (BERT) [1] is the state-of-the-art backbone for Natural Language Processing (NLP) tasks. It has an Embedding-Encoder-Decoder structure, where embeddings are learnable vectors for each token, the encoder is multiple stacked transformer blocks (embedding and encoder layers to be jointly referred to as '*representation-learner*'), and the decoder is a multi-layer perception (MLP). BERT pads the start of all text with a " *CLS*" token and the decoder only intakes the encoder output vector for this token, as it is assumed to have captured information from all other tokens in the sentence via the attention mechanism. Auxiliary non-text features can be concatenated to the *CLS* output vector before being fed into the decoder.

2.2 Feature Engineering

Text Representation Deep models intake tokens to map to embedding vectors. We crop or end-pad strings to the same length of 256 tokens and replaced tokens appearing less than 40 times in the corpus with 'UNK', as statistically good parameter estimation requires 40 observations. Our embedding hence contained 3993 tokens.

Perplexity: Indicates how well a probability model predicts a sample. High perplexity suggests that the text is unexpected or unlikely under the model.

Burstiness: Relates to the variance in sentence lengths within a text. Human language typically shows high burstiness due to dynamic and varied sentence structures.

String Length: Measures the number of tokens per string. Human-written texts often exhibit variable string lengths reflecting the flow of human thought, while machine-generated texts might be more uniform.

Unique Words Divided by Sentence Length: Assesses the richness and variability of vocabulary relative to the text length. Human authors generally employ a broader vocabulary in diverse contexts.

Domain: Label-less inference datas' domain derived from predictions of a 99% accuracy Light Gradient Boosting Machine model. This feature helps the model adapt to stylistic and thematic variations across different sources, enhancing its capability to differentiate between human and machine-generated texts.

2.3 Domain Adversarial Training of Neural Networks

Domain Adversarial Training of Neural Networks (DANN) [2] is a mechanism for improving primary model objectives when training data are from different sources. It coerces models to learn domain-neutral - as opposed to domain-specific - patterns from data by adding an auxiliary domain-classification decoder MLP, which intakes CLS's encoded vector. Domain predictions contribute to an auxiliary loss during training, whose gradients when back-propagated to representation-learner parameters are negated by a Gradient Reversal Layer (GRL) positioned between the encoder and domain-decoder. By feeding the representation-learner gradients opposite to improving domain classification, their weights are regularised to only capture shared patterns which benefits the classification task.

2.4 Two-systems learning within one model through Double Weighted Cross Entropy Loss

Weighted cross-entropy loss L_{WCE} (Equation 1) addresses label imbalance learning to optimize balanced accuracy. We further introduce a novel double weighted cross-entropy loss mechanism L_{DWCE} (Equation 2) whereby each domain's instances contribute to a domain-specific loss. This allows both domains to weight L_{WCE} based on their own distributions, with this two-systems loss framework addressing the issue of disrupting domain 1's 50-50 label balance when using a global L_{WCE} to improve domain 2's balanced accuracy. The two domain losses are weighted by domain distribution within each mini-batch when aggregated to the overall mini-batch loss. The mathematical formulae of losses are presented below:

$$L_{WCE}(y,\hat{y};w) = -\frac{2}{N}\sum_{i=1}^{N}[(1-w)\cdot y_i\cdot\log(\hat{y}_i)+w\cdot(1-y_i)\cdot\log(1-\hat{y}_i)], \tag{1}$$

where y and $\hat{y}$ are labels and predicted positive probabilities respectively, w is the prior distribution for the positive class labels; and:

$$L_{DWCE}(y_1,y_2,\hat{y}_1,\hat{y}_2,n_1,n_2;w_1,w_2,n) = 2[\frac{n_1}{n}\cdot L_{WCE}(y_1,\hat{y}_1,w_1)+\frac{n_2}{n}\cdot L_{WCE}(y_2,\hat{y}_2,w_2)], \tag{2}$$

where y_i, $\hat{y}_i$ and w_i are respectively the labels, predictions and prior distributions of sub-scripted domains. Batch size is represented by n, while n_1 and n_2 indicates the number of instances from a domain within each mini-batch.

2.5 Overall Model

Our final system comprises a single BERT model with an auxiliary domain decoder and loss, trained using the two-systems L_{DWCE} and DANN (Figure 1). The DANN loss also uses L_{WCE} with domain priors as weights. We begin training from randomised weights and early stop training based on average domain-balanced accuracy of the validation set. The final hyper-parameters are presented in Figure 2.

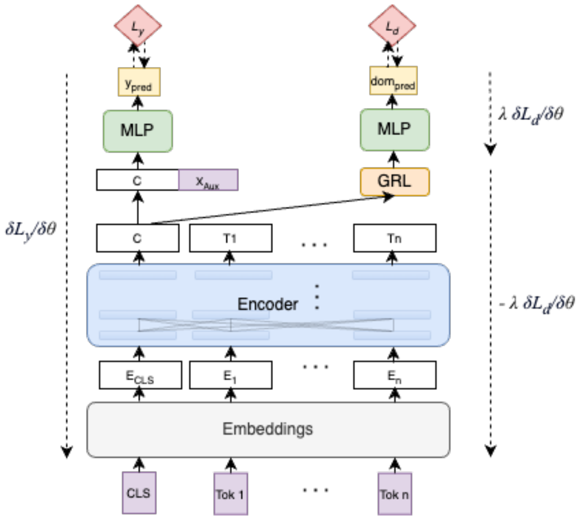

Figure 1: The BERT-DWCE-DANN system

Hyperparameter	Value
d_model	256
encoder_layers	8
decoder_y	3
decoder_dom	1
activation	ReLU
heads	8
batch_size	8
learning_rate	1e-5
patience	10
dropout	0.1

Figure 2: Model Hyperparameters

3 Discussion

In this section, we compare 4 models: 0R, Bert, Bert with DANN and Bert with both DANN and L_{DWCE}. The 4 models show the evolution of ideas from treating the task as a single domain NLP problem (hence the use of BERT rather than non-Deep models like SVM *[expr.2]* as BERT's representation learner generates more predictive signals compared to non-Deep's TF-IDF) to considering and tackling the domain and label imbalance as we compare the effect of adding different mechanisms so to test hypotheses. As the objective of the problem is to maximise accuracy for a completely balanced dataset over the two domains and labels (25% each), we evaluate the test set based on the **average of the two domains' balanced accuracy ($BAcc$)**. Model results are presented in Table 1.

Table 1: Balanced Accuracy of each experiment (Experiment ID labelled in 'Expr.' column)

Expr.	Model	Val Dom 1	Val Dom 2	Val Avg	Test Dom 1	Test Dom 2	Test Avg
1	0R	0.500	0.500	0.500	0.500	0.500	0.500
2	(SVM)	0.763	0.782	0.722	0.744	0.768	0.756
3	Bert L_{WCE}	0.731	0.847	0.789	0.751	0.852	0.795
4	(Bert L_{CE})	0.804	0.840	0.822	0.789	0.810	0.800
5	(Local Berts L_{WCE})	0.799	0.824	N/A	0.732	0.817	N/A
6	Bert DANN L_{WCE}	0.741	0.875	0.808	0.743	**0.863**	0.803
7	**Bert DANN** L_{DWCE}	0.833	0.886	0.860	**0.820**	0.854	**0.837**
8	(Bert L_{DWCE})	0.796	0.846	0.821	0.795	0.851	0.823
9	(Bert DANN L_{DWCE} No Auxiliary Features)	0.763	0.808	0.763	0.812	0.785	0.787

3.1 Observation 1: More data preferred over domain-only training data

Given the success of predicting domains with 99% accuracy, experiments were conducted to train two local models using only data from each domain, and set up the inference pipeline to feed inputs into the model according to the predicted domain. The rationale was based on observing BERT trained with both unweighted cross-entropy loss (L_{CE}) and L_{WCE} *[expr.3, 4]* producing inferior test BAcc performance on domain 1 data compared to domain 2. This led to the hypothesis that the data of the two domains had substantially different patterns, and hence benefits of training with more data were outweighed by poor learning due to cross-domain inflicted noise for the minority domain. The two local models experiment on Bert demonstrated adverse effects compared to a global model (-3.8% and -3.5% respectively for each domain *[expr.5 vs 3]*, suggesting BERT favoured more training instances in the aforementioned trade-off.

3.2 Observation 2: Potential domain-difference is not the primary adverse effect

Despite observation 1, we continue to hypothesise the existence of domain-specific patterns having a negative impact on training as the fundamental under-performance of the minority domain persisted. We use the DANN training mechanism to test our hypothesis: if DANN improved model performance and narrowed the gap between the two classes, then we conclude our hypothesised negative impacts existed (refer to subsection 2.3 for rationale).

Empirical results *[expr.6]* did not show any significant difference for domain 1, but yielded 0.9% improvement in domain 2 to achieve its best BAcc over all experiments, demonstrating the majority class to benefit from domain-neutral representation-learning. This result nonetheless only furthered the gap in domain performance, hence suggesting that control over more dominant adverse effects was required before a conclusive conclusion regarding cross-domain adverse effects could be determined.

3.3 Observation 3: Double loss resolves label domain dual imbalance problem

In the original Bert L_{CE} and L_{WCE} *[expr.3 vs 4]* comparison, we observe that domain 2 performance increases at the cost of domain 1 performance when using L_{WCE}. This leads us to hypothesise that as domain 1 is balanced in labels but domain 2 has a 88%-11% split and is the majority domain, using the global label prior of 78%-22% to weight L_{WCE} upsets domain 1's label balance and hence worsens its learning. This leads us to experiment with splitting the two losses such that domain 1 instances contribute to a balanced cross-entropy loss while having domain 2 to use L_{WCE} with its own prior. Empirical results *[expr.7]* on the test set demonstrated a significant boost for domain 1 of 7.7% BAcc and an overall 3.7% performance gain in the domain averaged BAcc. This confirms the hypothesis that the primary adverse effect hindering performance was global weighting or non-weighting of L_{WCE} and L_{CE}, respectively, was trading off domain 2's performance gain with domain 1's.

Having controlled the primary adverse effect, we returned to evaluating whether domain-specific patterns were negatively impacting performance. Empirical evidence *[expr.8]* shows a 2.5% and 0.3% decrease in performance of domain 1 and 2 BAcc respectively, when DANN is removed from the final model. This suggests that DANN is regularising against learning domain-specific representations to improve performance, and hence domain-specific patterns have a negative impact which is especially significant for the minority class. Also observing our best model with auxiliary features removed *[expr.9]* from decoder input, we witness performance decline in both domains (5% overall). This suggests substantial differences in perplexity, burstiness etc. in human and AI text, but also possible differences of these characteristics in domains as they could have been benefiting from the conditioning effect of the auxiliary feature 'domain'; however, this cannot be conclusive without further experiments.

4 Limitations and Conclusion

In this work, we developed a system for predicting human and AI-generated text with data from two source domains. Our final system used a BERT model with domain adversarial training and a novel domain-separate double weighted loss to achieve a 4.2% domain averaged balanced accuracy improvement from our baseline, 33.7% from 0R and 85.3% absolute accuracy on the public Kaggle dataset. Our experiments showed that the double weighted cross-entropy loss two-systems learning mechanism effectively addressed the primary challenge of this problem which was that different label distributions in the two domains caused trade-offs between domain performances when balancing labels globally; and the existence of negative effects of domain discrepancy which could be limited by DANN.

However, limitations of this framework calls for further investigation: the double loss system has only been experimented for this domain-label distribution; its effectiveness on other distribution combinations needs to be experimented with. The scalability of this framework to more domains (2+ loss functions) should also be investigated, particularly if some domains have extremely few observations.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: `1810.04805 [cs.CL]`.

[2] Y. Ganin, E. Ustinova, H. Ajakan, *et al.*, *Domain-adversarial training of neural networks*, 2016. arXiv: `1505.07818 [stat.ML]`.