GLOBAL RESEARCH IMMERSION PROGRAM FOR YOUNG SCIENTISTS

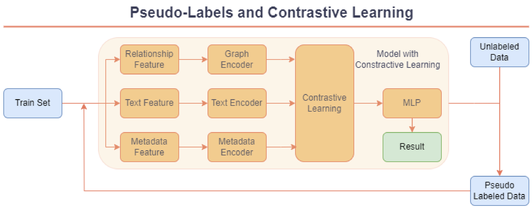
Semi-supervised Bot Detection: Leveraging Pseudo-Labels and Contrastive Learning

Jiayi Li[1], Tianyang Wu[2]
[1]University of Sydney, [2]University of Illinois (UIUC)

Abstract

The detection of Twitter bots is crucial to combat misinformation, election interference, and propaganda. Advanced bots disguise themselves by mimicking genuine users, making detection challenging. To address this, BotMoE [2] is proposed, a framework utilizing metadata, textual content, and network structure to improve detection. BotMoE includes a community-aware Mixture-of-Experts (MoE) layer for domain generalization across different Twitter communities. It uses modal-specific encoders and an expert fusion layer to combine these modalities and measure user information consistency. Experiments show BotMoE's superior performance in detecting advanced, evasive bots and its ability to generalize to new user communities. In this project, we aimed to improve BotMoE's performance to a higher level by adding contrastive learning and semi-supervised learning to the existing model.

Overview



Our approach involves three distinct sub-models designed to process various data modalities: a graph structure encoder, a text encoder, and a metadata encoder. We utilize contrastive learning to integrate the outputs from these encoders, thereby ensuring feature consistency and achieving robust classification results.

The initial training phase employs labeled data to develop the primary model. Subsequently, this trained model is used to infer pseudo-labels for the unlabeled dataset. A selected portion of these pseudo-labeled samples is then combined with the original labeled data to refine the model through additional training. [1]

Introduction

Twitter bots, automated accounts, pose significant threats by spreading misinformation, deepening divides, promoting conspiracies, and influencing elections. Detection efforts have evolved from feature-based to text-based and graph-based models due to bot operators manipulating metadata and content. Despite progress, challenges remain with bots manipulating multi-modal features and existing in diverse communities. Existing detection models often fail to address these complexities. BotMoE, a novel framework, employs a community-aware mixture-of-experts architecture, leveraging multi-modal user information to tackle feature manipulation and diverse community challenges. Extensive experiments show BotMoE's effectiveness and generalization capabilities.

Methods

Contrastive Learning

The contrastive learning component in the model is designed to distinguish between similar and dissimilar feature representations to improve the consistency of the multi-modal feature representations [6],[4]. This is achieved through the following steps:

1. Example Construction: The model considers the attention outputs from one account are similar, thus positive examples:

$$\text{positive\_examples} = \text{attention\_layer\_outputs}$$

Negative examples are created by pairing the features of one account with features from another account in that batch:

$$\text{negative\_examples} = \text{positive\_examples}[\text{randperm}(\text{batch\_size})]$$

2. Similarity Calculation: The cosine similarity [5] between each example pair is computed:

$$\text{positive\_similarity} = \text{cosine\_similarity}(\text{positive\_pairs})$$

The diagonal elements of the similarity matrix represent the similarity of each positive example with itself:

$$\text{positive\_loss} = 1 - \text{positive\_similarity.diag}().\text{mean}()$$

Similarly,

$$\text{negative\_similarity} = \text{cosine\_similarity}(\text{negative\_pairs})$$

$$\text{negative\_loss} = \text{negative\_similarity.mean}()$$

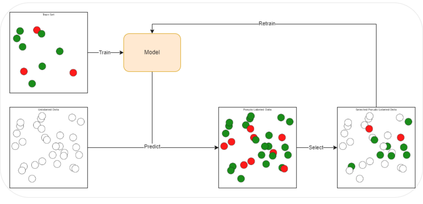
3. Contrastive Loss Calculation: The total contrastive loss is computed as the sum of the positive loss and the negative loss:

$$\text{contrastive\_loss} = \text{positive\_loss} + \text{negative\_loss}$$

This contrastive loss is then scaled to match the order of magnitude of the export loss and added to the overall loss function. Thus, the feature output from the attention layer will tend to gather similar examples and provide better consistency.

Methods (cont.)

Pseudo-labeling



The following process is followed for pseudo-labeling and training:

1. Train the initial model using the labeled dataset.
2. Employ the initial model to predict pseudo-labels for the unlabeled dataset, selecting appropriate samples based on confidence and uncertainty. Specifically, we assume that samples with confidence greater than the 80th percentile and uncertainty less than the 20th percentile are meaningful. [3]
3. Aggregate the original labeled data and the pseudo-labeled data for the next iteration, continuing this process until all unlabeled data has been utilized.

To prevent the model from accumulating errors, we retrain the model from scratch at each iteration.

Discussion

After experimentation, the inability of the model to improve its accuracy is not because the training set is not large enough, it is because it does not contain informative quality data. Therefore even if the correctness of the pseudo-labelling can be guaranteed, it does not enable the model to get more meaningful data for training.

References

References

[1] Paola Cascante-Bonilla et al. "Curriculum Labeling: Revisiting Pseudo-Labeling for Semi-Supervised Learning". In: *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*. https://ojs.aaai.org/index.php/AAAI/article/view/16200. 2021.

[2] Yuhan Liu et al. "BotMoE: Twitter Bot Detection with Community-Aware Mixtures of Modal-Specific Experts". In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. ACM, 2023.

[3] Nabeel Seedat et al. "You can't handle the (dirty) truth: Data-centric insights improve pseudo-labeling". In: *Journal of Data-centric Machine Learning Research* (2024). https://openreview.net/forum?id=2tBwcT9z55.

[4] Harald Steck, Chaitanya Ekanadham, and Nathan Kallus. "Is Cosine-Similarity of Embeddings Really About Similarity?" In: *arXiv preprint arXiv:2403.05440* (2024). https://arxiv.org/abs/2403.05440.

[5] Peipei Xia, Li Zhang, and Fanzhang Li. "Learning similarity with cosine similarity ensemble". In: *Information Sciences* 307 (2015), pp. 39–52. DOI: 10.1016/j.ins.2015.02.024. URL: %5Curl%7Bhttps://doi.org/10.1016/j.ins.2015.02.024%7D.

[6] Zijuan Zhao et al. "Heterogeneous Graph Contrastive Learning with Augmentation Graph". In: *IEEE Transactions on Artificial Intelligence* XX.XX (2024). https://doi.org/10.1109/TAI.2024.3400751, pp. 1–10.