Detecting Offensive Language in Tweets with Attention-fused Text and User Embeddings

Viha Gupta,[1] **Casey Primel**[1]

[1] Dept. of Computer Science
vg2237@nyu.edu, ctp219@nyu.edu

Abstract

Transformer-based models have come to dominate natural language processing tasks including semantic evaluation of social media content. Recent research demonstrates that the performance of such models can be improved by incorporating information about the online communities where that content is produced. For our project, we experiment with a novel method for incorporating community structure features and text features to classify offensive language via attention mechanisms and positional encoding. In comparison to existing published solutions, we incorporate a "corrected" procedure for computing attention on graph-structured data to produce our user embeddings and a better performing pre-trained language model for our text embeddings. Our model achieves an F1 score of **FINAL SCORE** outperforming our baseline model. Our code is available at https://github.com/guptaviha/GF-OLD.

Introduction

Content moderation is a pressing concern for social media platforms. For over a decade, automated systems have been employed to identify abusive behaviors and offensive language in user-generated content. Early systems relied on hand-crafted rules to extract linguistic features from textual data. Such rule-based systems have largely been replaced by neural network-based systems that can identify more subtle patterns in users' linguistic behaviors. However, research has shown that textual features are not always sufficient for identifying the nature of user-generated content. The rise of social media and the corollary ability to query the social graphs formed by users' activity has opened up a possible data source for constructing the requisite context. The remaining question is how to most effectively provide such context from the available data. Our project focuses on one solution in the context of Twitter: the fusion of text and user embeddings via attention mechanisms.

Related work

Recent work exploring the application of deep learning for offensive language detection on Twitter can be divided into two different approaches. The first approach focuses solely on text data (e.g., tweets) and the use of pre-trained language models fine-tuned for this domain-specific task. (Liu, Li, and Zou 2019). DISCUSS TWEETEVAL

The second approach combines text data with information about users and community structure. Recent exemplars of this approach have used graph neural networks to learn the features of the social graph: Mishra, Yannakoudakis, and Shutova (2019) constructs a heterogeneous graph of users and tweets and applies a graph convolutional network to learn its features. Miao et al. (2022) begins by independently extracting text and user embeddings and then fusing them via attention mechanisms. The latter serves as the starting point for our project. EXPAND BOTH OF THESE.

Methodology

DESCRIBE OBJECTIVES.

Dataset and preprocessing

For this project, we use the English-language dataset collected, labeleled and made publicly available on GitHub by Miao et al. (2022).[1] The data was obtained via the Twitter API[2] and contains data from 1,260 users and 12,780 of their tweets posted between Janauary 2018 and March 2021. The dataset also captures the social community in terms of the follower-network between these users with 8,877 such relationships. Unlike other Twitter datasets, Miao et al. (2022) construct a dataset to preserve community structures and better reflect real-world distribution of offensive language. They accomplish this by constructing a lexicon of topic-related words rather than offensive words from which they then query for users and their first- and second-order friends whose tweets are also collected. The result is a dataset where the ratio of offensive tweets is 7.90%, i.e., 1,009 out of the 12,780 total.

To represent the social graph, the users and relationship data are used to construct a graph structure with the users represented by nodes and the follower relationships represented by edges. Drawing on the work of Mishra, Yannakoudakis, and Shutova (2019), Miao et al. (2022) use the users' historical behavior to represent the propensity of the

[1] https://github.com/mzx4936/GF-OLD-Dataset
[2] https://developer.twitter.com/en/docs/twitter-api

user to post offensive language. The number of tweets labelled non-offensive and offensive in the training data are used to describe users' historical behavior, i.e., the number of non-offensive tweets and the number of offensive tweets from the training data made by a user are each set as node attributes.

Several text preprocessing steps are taken to prepare the raw text data: emojis are converted to words with similar meanings using third-party libraries[3], URLs replaced with "*http*", hashtags segmented into phrases, and text like user references, dates and email addresses are converted into uniform placeholders, e.g., "*@anonymous2134*" becomes "*<user>*".

Model architecture

We will be following Miao et al. (2022) in using a graph attention network for learning user embeddings and attention mechanisms for fusing user features with tweet features; however, rather than using text embeddings from BERT, we will use the RoBERTa-based model pre-trained on tweets and fine-tuned for offensive language detection (Barbieri et al. 2020).[4]

All the models discussed here use Adam optimizer (Kingma and Ba 2014) and focal loss function (Lin et al. 2017), a modified cross entropy loss designed for dealing with class imbalance. It introduces a scaling factor $(1-p_t)^\gamma$ such that setting $\gamma > 0$ reduces the relative loss for easy examples and puts more emphasis on hard to classify examples. The function can be written

$$FL(p_t) = \alpha \cdot (1-p_t)^\gamma \log p_t$$

where $p_t = \begin{cases} p, & \text{if } y = 1 \\ 1-p, & \text{otherwise} \end{cases}$, i.e., the magnitude of the estimated probabilty irrespective of the class label. In the special case where $\gamma = 0$, the focal loss function simplifies to cross entropy loss.

Transformer-based language models BERT AND DIFFERENCE BETWEEN BERT AND ROBERTA.

Graph attention networks Graph attention networks (GAT) are an attention-based graph neural network (GNN) architecture for performing node classification of graph-structured data introduced by Veličković et al. (2017). Here, attention is used as a means of neighborhood aggregation wherein the hidden representation of each node in the graph is computed by attending over all its neighbors and selecting those which are most relevant. Mathematically, GAT computes a score of every edge in the graph which indicates the importance of the features of a given node's neighbor to the given node, $e(\mathbf{h}_i, \mathbf{h}_j) = LeakyReLU(\boldsymbol{a}^T \cdot [\mathbf{W}\mathbf{h}_i || \mathbf{W}\mathbf{h}_j])$, where both $\boldsymbol{a}$ and $\mathbf{W}$ are learned. From here, it computes a learned weighted average of the representations of a given node's neighbors followed by a nonlinearity.

As noted by Brody, Alon, and Yahav (2021), the expressiveness of GAT as formulated by Veličković et al. (2017) is constrained because the ranking of node importance computed as part of its scoring function ends up being shared by all nodes in the graph. Brody, Alon, and Yahav (2021) refer to this as *static attention* and demonstrate how static attention prevents GAT from approximating even very simple functions. To remedy this, Brody, Alon, and Yahav (2021) suggest a simple reordering of the operations in GAT resulting in what they refer to as GATv2. The reordering separates out the application of $\boldsymbol{a}$ and $\mathbf{W}$ in the scoring function so that they do not collapse into a single linear layer: $e(\mathbf{h}_i, \mathbf{h}_j) = \boldsymbol{a}^T LeakyReLU(\mathbf{W}[\mathbf{h}_i || \mathbf{h}_j])$.

Miao et al. (2022) use GAT to derive their user embeddings prior to performing attention fusion. Because GAT exhibits static attention, the user embeddings derived emphasize global community structure and de-emphasize local structures within the social graph. We hypothesize that substituting GAT with GATv2 will allow the user embeddings to better express the local structures of the social graph.

Attention fusion Attention mechanisms are used to fuse user and text embeddings derived from the graph attention network and the language model. First, the user embeddings corresponding to the current batch's text samples are selected and, then, appended to the text embeddings. The combined embeddings are positionally encoded and input into a multiheaded attention module using scaled dot-product attention as described by Vaswani et al. (2017) after which a residual connection and layer normalization are applied.

Training setup

DESCRIBE TRAINING SETUP: SCRIPTS AND HARDWARE.

Experiments

DESCRIBE EXPERIMENTS RUN AND RATIONALE.

Results

For our final results, ...

Discussion

Our final model is able to achieve a test accuracy of SCORE and an F1 score of SCORE on Miao et al. (2022) Twitter dataset surpassing our baseline by XYZ and XYZ, respectively. There are several avenues for further exploration and possible improvement that we can identify. The first would be a more thorough comparison of models that incorporated hyperparameter tuning on a per-model basis, especially in regard to the application of different learning rates and regularization. The second is to revisit the construction of the social graph and the derivation of the user embeddings to evaluate how well they are capturing community structure. In particular, one could use non-parametric methods for community detection as a baseline against which to measure how effectively the graph attention layer as capturing community features.

[3]emoji (https://github.com/carpedm20/emoji) and ekphrasis (https://github.com/cbaziotis/ekphrasis).

[4]https://huggingface.co/cardiffnlp/twitter-roberta-base-offensive

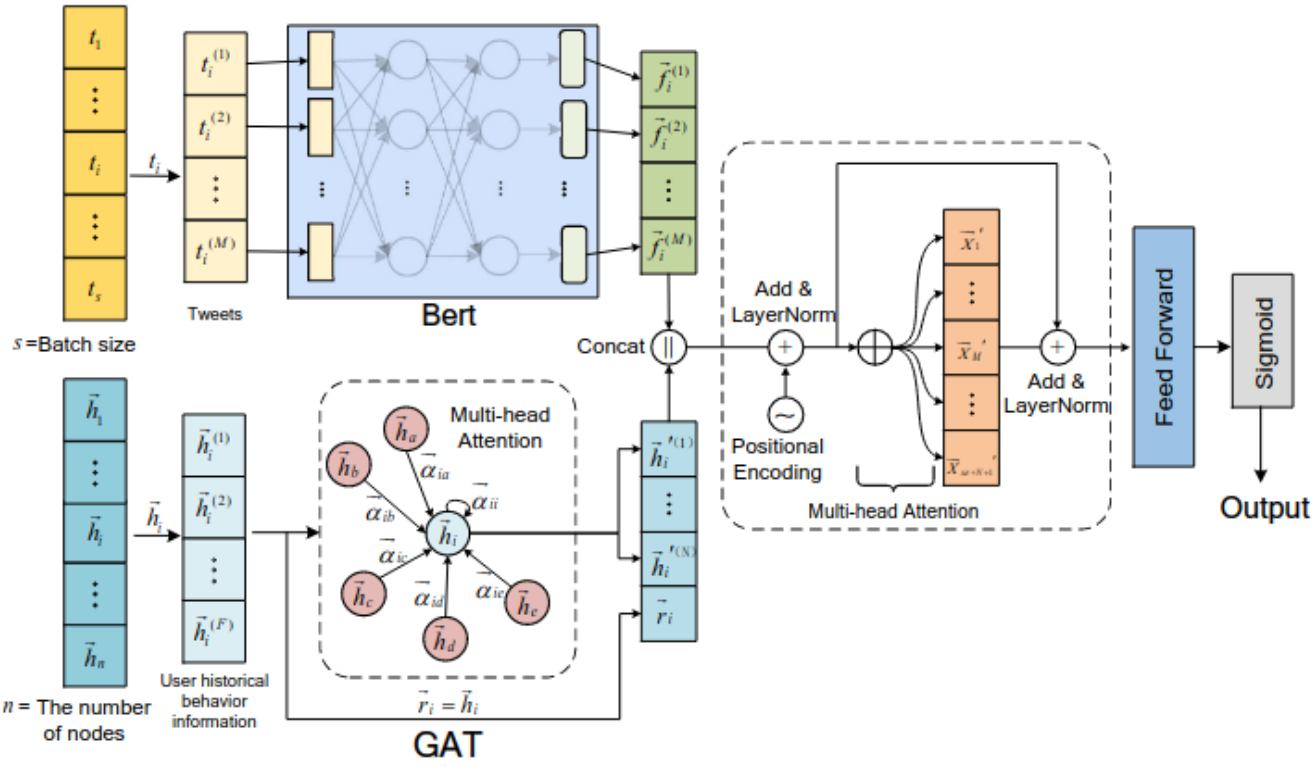

Figure 1: End-to-end offensive language detection model (Miao et al. 2022).

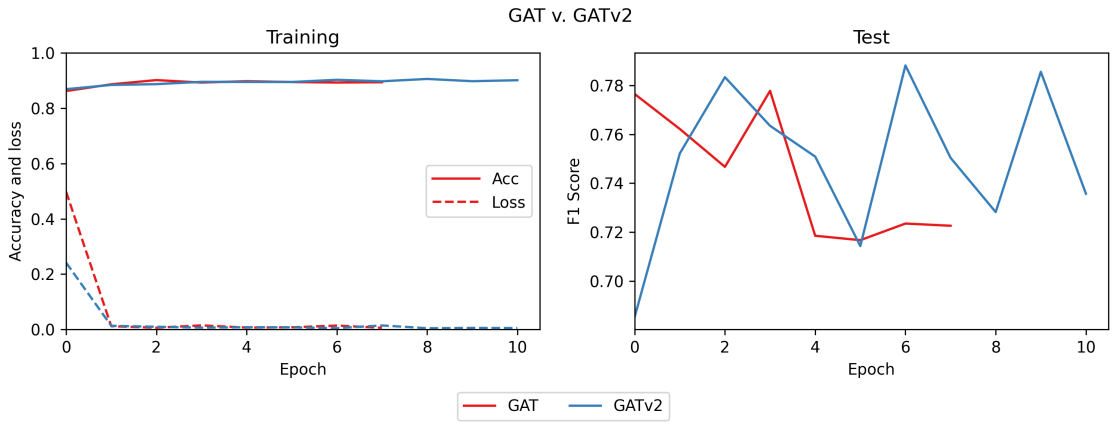

Figure 2: Comparison of model training accuracy, training loss and test F1 score for GAT and GATv2.

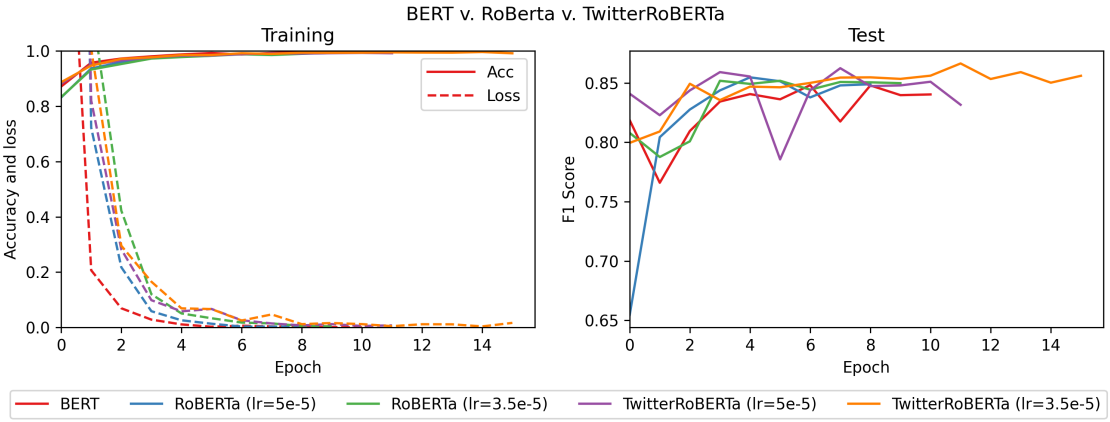

Figure 3: Comparison of model training accuracy, training loss and test F1 score for BERT, RoBERTa and TwitterRoBERTa.

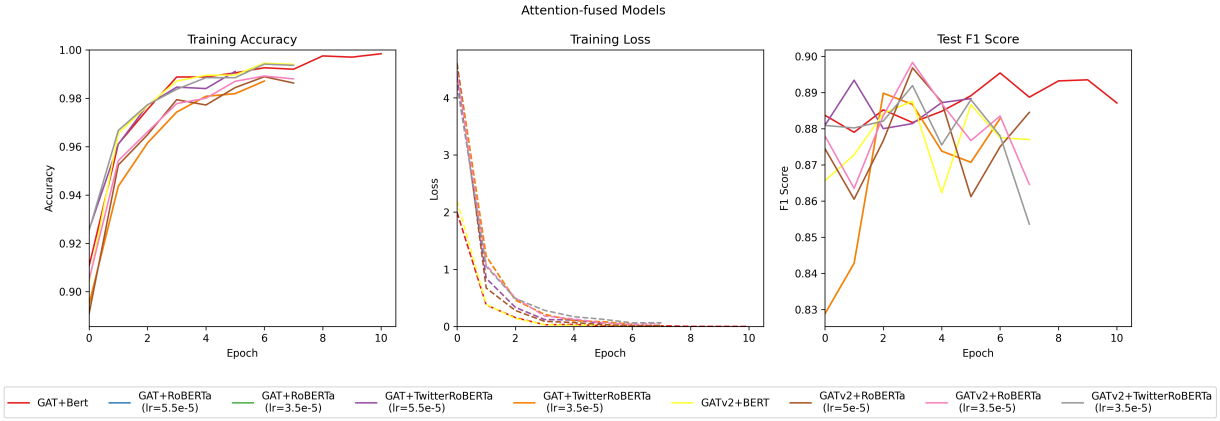

Figure 4: Comparison of model training accuracy, training loss and test F1 score for all attention-fused models.

References

Barbieri, F.; Camacho-Collados, J.; Neves, L.; and Espinosa-Anke, L. 2020. TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification.

Brody, S.; Alon, U.; and Yahav, E. 2021. How Attentive are Graph Attention Networks?

Kingma, D. P.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Liu, P.; Li, W.; and Zou, L. 2019. NULI at SemEval-2019 Task 6: Transfer Learning for Offensive Language Detection using Bidirectional Transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, 87–91. Minneapolis, Minnesota, USA: Association for Computational Linguistics.

Miao, Z.; Chen, X.; Wang, H.; Tang, R.; Yang, Z.; and Tang, W. 2022. Detecting Offensive Language on Social Networks: An End-to-end Detection Method based on Graph Attention Networks.

Mishra, P.; Yannakoudakis, H.; and Shutova, E. 2019. Tackling Online Abuse: A Survey of Automated Abuse Detection Methods.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2017. Graph Attention Networks.