Now You See Me:
Vision-enhanced BERT for obfuscated text abuse detection

Stanford CS224N Custom Project

Dylan Zhou
Department of Computer Science
Stanford University
dylanzhou@stanford.edu

Abstract

Language models today have demonstrated remarkable capabilities and revolutionized our understanding of machine intelligence. Yet even still, transformations to model inputs—even those seemingly insignificant to the human eye—have the power to completely confuse the model and derail its outputs. We refer to these transformations as *text obfuscation*, and in this project, we explore methods to patch the vulnerabilities presented by text obfuscation in a BERT-based abusive text classification model. In particular, we experiment with combining inputs from Vision Transformer models to augment BERT's ability to "see" and correctly classify a wider range of text inputs. Our methods show a performance improvement of **+7.8%** in accuracy and **+10.4%** in precision on a benchmark dataset for obfuscated hate speech detection.

1 Key Information to include

- Mentor: Shijia Yang
- External Collaborators (if you have any): None
- Sharing project: No

2 Introduction

Consider the interaction with chatGPT shown in Figure 1. While we can easily make out the text—which depicts a stylized version of the words "GITHUB ACTIONS"—one of today's most powerful language models is completely unable to decipher it, instead guessing that it says "HELLO WORLD".

This result likely poses many kinds of problems to applications of language models in a variety of situations, but we will focus on one very specific problem in this area: adversarial inputs to a language model used for content moderation. For example, suppose we run a social media platform and want to minimize the amount of hate speech posts on our platform. We maintain a language model that classifies each post as `hate_speech` or `not_hate_speech` and removes the `hate_speech` posts. While it could have very high accuracy on regular non-obfuscated posts, adversarial users could bypass the model by obfuscating their text in some way that makes it unreadable to the model but easily decipherable by other human users.

In this project, we finetune a BERT model on hate speech data to use as a toy example of a content moderation model. We create a simple custom text-obfuscation library that turns ASCII characters into similar-looking non-ASCII characters that are untokenizable by our BERT classifier and we demonstrate the model's shortcomings on a hate speech dataset obfuscated using our library. Then we explore ways to recover some performance of the BERT classifier by first rendering the obfuscated text as an image, then passing it through a finetuned Vision Transformer (ViT), and then feeding signal from the ViT to our BERT classifier for classification, with the hope that the Vision Transformer acts as "eyes" for our BERT model and allows it to more accurately read the obfuscated text.

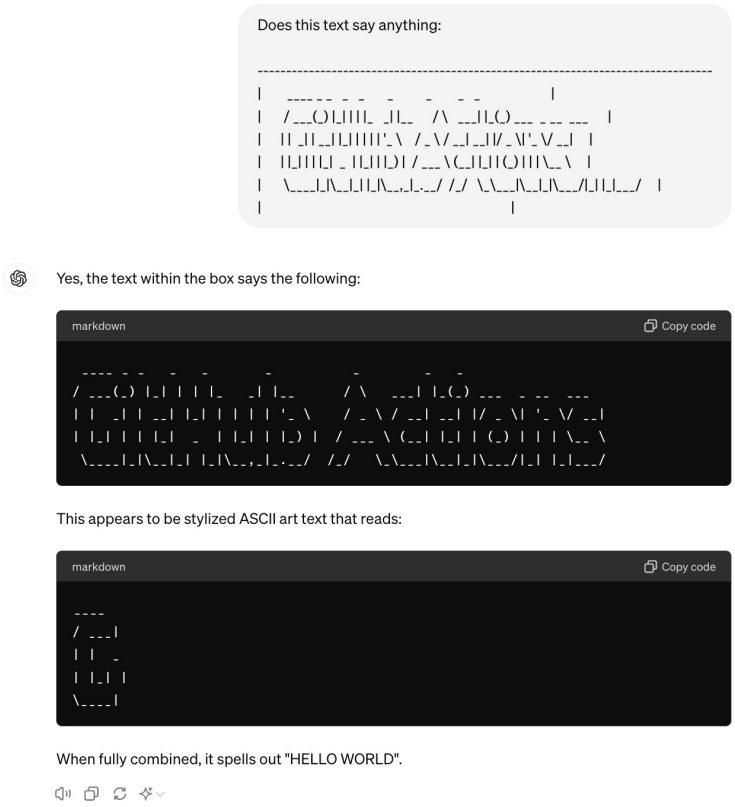

Figure 1: When asked to read an ASCII-art-rendering of the text "GITHUB ACTIONS", chatGPT mistakes it for "HELLO WORLD", demonstrating the text obfuscation problem for language models.

For this project, we restrict our attempts at performance recovery by never re-training the finetuned BERT model and never training our hybrid ViT-BERT model end-to-end. This choice is to simulate the lengthy training times an industry-level model likely has for training their base models—which could be hundreds of billions of parameters large and take weeks or months to retrain—and to preserve the good performance already inherent in those base models, which are already optimized for abuse detection. Instead, what we want to do is to demonstrate a way to build on top of that foundation and add additional functionality (the ability to read obfuscated text) with minimal changes to the underlying model.

In the end, we show that by passing signals from a finetuned ViT model to our fixed BERT classifier, we are able to improve upon the accuracy and precision over the original "blind" BERT model by 7.8% and 10.4% respectively when classifying obfuscated hate speech text.

3 Related Work

3.1 Vision Transformer

In their paper *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, Dosovitskiy et al. (2021) introduce a novel approach to image classification by adapting the Transformer architecture, traditionally used for natural language processing, to visual data. Instead of using convolutions, the Vision Transformer (ViT) divides an image into a sequence of fixed-size patches, linearly embeds each patch, and feeds these embeddings into a standard Transformer encoder. This approach allows the model to capture long-range dependencies and global context more effectively. The paper demonstrates that ViT can achieve state-of-the-art performance on image classification benchmarks with significantly fewer computational resources compared to traditional convolutional neural networks, especially when pre-trained on large datasets.

For our problem we hypothesized that because ViT and BERT share the transformer architecture, it would be easier to align their representations of information and easier to pass information between the two models than it otherwise would be for two models that do not share similar architectural components. With this line of reasoning, we chose to finetune ViT models to read rendered images of text and pass their representation of those images to the BERT classifier for inference.

3.2 Contrastive Learning Methods

To align the representation of information between the two models, we took inspiration from contrastive learning methods, one being CLIP and another being simCLR.

CLIP (Contrastive Language-Image Pre-training), developed by Radford et al. (2021) is a framework that learns visual concepts from natural language supervision by leveraging a contrastive loss. The model consists of separate image and text encoders, which project images and their corresponding textual descriptions into a shared embedding space. The contrastive loss function is used to bring the embeddings of matching image-text pairs closer together while pushing non-matching pairs apart. This training paradigm enables CLIP to perform zero-shot transfer learning, demonstrating impressive performance across various visual tasks without requiring task-specific fine-tuning. The use of contrastive loss is central to CLIP's ability to align visual and textual representations effectively.

We had a related problem, where we wanted to align separate image encoders (from ViT) and text encoders (from BERT) so that the image encoders would represent images of text closely to how the text encoders would represent the original text. We ended up using some of their techniques, such as projecting into a shared embedding space.

Another paper we took inspiration from is simCLR. Chen et al. (2020) introduce a straightforward and effective approach to unsupervised visual representation learning using contrastive learning in their paper *A Simple Framework for Contrastive Learning of Visual Representations*. The framework consists of four main components: data augmentation, a neural network base encoder, a projection head, and a contrastive loss function. The key idea is to maximize the agreement between different augmented views of the same image in the latent space while minimizing the agreement between views of different images. This is achieved by applying various augmentations to the input images, passing them through the encoder and projection head, and then using a contrastive loss to train the model. SimCLR shows that strong data augmentation, a large batch size, and a carefully designed projection head are critical for achieving high-quality visual representations that rival or surpass those learned with supervised methods on various downstream tasks.

The elements that we learned from here were their loss function, which they introduce as *NT-Xent* (the normalized temperature-scaled cross entropy loss) and which we will describe in further detail below, and the projection heads—both elements worked well empirically for our use case.

4 Approach

4.1 Baselines

First we finetuned a BERT model for the hate speech classification task using a pretrained checkpoint from Huggingface (google-bert/bert-base-uncased) (model developed by Devlin et al. (2019)) and data from Mody et al. (2023). We achieved 83.8% accuracy and 85.0% precision on a non-obfuscated test set. This model became our fixed base model for hate speech classification.

Next, we created a custom obfuscation library that maps uppercase and lowercase ASCII characters in a given string into non-ASCII characters (see Figure 2 for an example).

```
Original: Here is an example of text to be obfuscated.
Obfuscated: Ḩęȓę iş αή ęϰαṃṗlę ȍʃ ṫęϰṫ ṫȍ ҍę ȍҍʃυşçαṫęd.
```

Figure 2: Example use of text obfuscation library.

After applying obfuscation to our test dataset, we observed 49.4% accuracy and 49.4% precision from our BERT classifier.

We then rendered all of the text datasets (obfuscated and non-obfuscated) as images in preparation for our experiments (see Figure 3).

Figure 3: A 4x4 grid of examples of our rendered obfuscated test set. Images are 224 by 224 pixels to match the input size expected for ViT models.

4.2 Vision-enhanced BERT

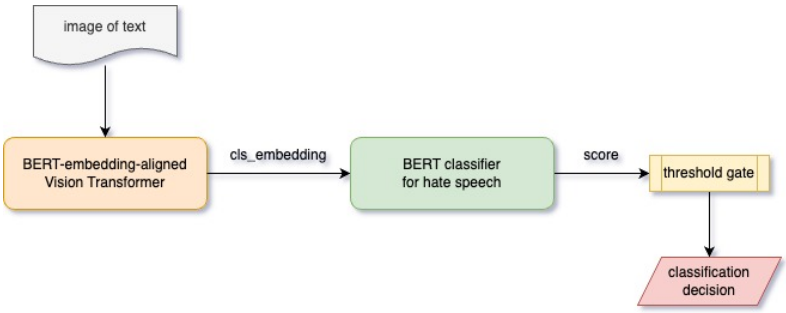

Figure 4: Vision-enhanced BERT architecture diagram.

Our vision-enhanced BERT classifier consists of two main parts. First, a ViT model takes in an input image of text and generates embedding representation of the text. Second, our already-finetuned BERT classifier takes the ViT model's embedding and outputs a score, which we pass through a threshold gate to determine the final classification decision.

More details will follow in the next section, but from a high level, we used contrastive learning to steer the embeddings generated by the ViT model to be aligned with the embeddings generated by our

BERT classifier (*i.e.*, for a given text input and its corresponding rendered image, we tried to have the ViT model generate an embedding for the image that was as close as possible to the BERT model's embedding for the original text). We hoped that in doing so, the ViT model would embed obfuscated text close to its non-obfuscated version and generate embeddings that are not only interpretable but also still correctly classifiable by the BERT model.

5 Experiments

5.1 Data

We use a balanced Hate Speech Dataset from Mody et al. (2023), which contains short text comments and labels designating each comment as hate speech or not. While the dataset contains 700,000 items, we shuffle and use only 8000 for training, 1000 for validation, and 1000 for testing. Due to the balanced nature of the dataset, roughly half of each set is hate speech and the other half is not.

For the test set, we use our custom obfuscation library to generated an obfuscated copy. We also use the Python Imaging Library (PIL) to generate rendered 224x224 pixel images of each text datapoint (see Figure 3 for an example). We generate an obfuscated version of only the test set because we want to simulate a industry production setting, where we may not be aware of an adversarial user's obfuscation methods, so we cannot augment our training data using the same obfuscation techniques.

5.2 Evaluation method

We use accuracy and precision metrics to evaluate our model. We purposely exclude recall, another commonly used metric, because it is easy to score perfectly on recall by labeling all datapoints as hate speech—in fact, this is how the baseline BERT classifier behaves on the obfuscated dataset, and we do not want to consider this a good model.

We set out to compare our vision-enhanced BERT model to two baseline models. The first point of comparison is the BERT classifier's performance on the obfuscated test set. The second point of comparison is the BERT classifier's performance on the obfuscated test set pre-processed by an off-the-shelf optical character recognition (OCR) tool that attempts to deobfuscate the text into ASCII characters. We use `pytesseract`, a commonly used Python OCR tool for this step. We will compare the performance of these models with our vision-enhanced BERT model in the Results section below.

5.3 Experimental details

5.3.1 Initial attempts

In our very first experiment, we tried passing the embeddings from a pretrained ViT with no finetuning straight into our BERT classifier. We observed that everything was classified as hate speech, which matched the first baseline's performance of 49.4% on both accuracy and recall.

We hypothesized that if we could get the embeddings from the ViT model for a given image to line up with the BERT model's embeddings for the corresponding original text, then we would have some improvement gain. We implemented our first attempt at aligning the embeddings by generating BERT embeddings for the training dataset and finetuning the ViT model with a mean-squared-error (MSE) loss between the ViT model's CLS embeddings and the BERT model's text embeddings. During training, we noticed that the loss was barely changing, so we decided to take another look at our data.

5.3.2 Data rendering improvement

At first, we had a more basic text-to-image rendering method, which simply rendered the images in non-standard rectangles where the text was all in one line. When passing these to the ViT model, we had to transform them into 224x224 pixel images which warped the text, making it hard to read. This could explain our ViT model's stagnant loss during finetuning.

We tried to remedy the problem by adding padding. This did not fix the stagnant loss problem. Taking another look at this data, we saw that while padding resolved the warping issue, there was now a scaling issue, where long text comments would still be in one line but they would be rendered extremely small and the text was often illegible. Furthermore, without any data augmentation here,

we guessed that observing text at different scales might confuse the model when finetuning on this data.

Finally, we implemented our current version of the rendering step, which uses a consistent font size and adds new lines where necessary so that the text is rendered legibly and at a consistent scale in a 224x224 pixel image (see Figure 3).

5.3.3 Projection heads and new loss function

With this new data, we unfortunately still observed no progress on the loss, so we looked to other papers involving contrastive learning, such as CLIP and simCLR, for inspiration.

Our first change was changing our loss function from MSE to NT-Xent, which simCLR showed performed well for contrastive learning. For an in-depth explanation of NT-Xent loss, see Chen et al. (2020).

Another new technique we implemented was adding a projection head layer that would project both the ViT CLS embeddings and the BERT text embeddings to a smaller dimension and calculate the loss between those smaller vectors. Supposedly, having a smaller projection layer makes it easier for two representations to become aligned.

The default size for both the ViT and BERT embeddings is 768, and after trying various values for a smaller projection layer, ranging from 128 to 512, we observed some progress in the loss for a projection dimension of 512.

5.3.4 Extended training, learning rate scheduling, and score mode

Thus far we had been training all of our models for 5 epochs but with various learning rates and weight decay values. We finally found a configuration that seemed to work, with the 512-dimensional embedding projection space, the NT-Xent loss function, and a learning rate of 1e-3. For this model, we continued training for 30 epochs with an additional learning rate exponential decay with a gamma value of 0.9. At this point the loss had decreased from 1.3865 on the validation set to 1.2158 and roughly stabilized.

We ran this on the test set, and yet we still saw no progress made. At this point, we re-examined how we got the final classification decision and looked into the logits themselves. We made a modification for the BERT model to output a score between 0 and 1 for the final classification decision and we wrote a function to grid-search for the optimal threshold for accuracy and precision. Using this new method, we were able to achieve positive results!

5.4 Results

Given that we modified the final model to use a score mode plus threshold gating, we modified the baselines to use the same method to make a fair comparison. Results reported here reflect the best results obtained using the score mode (which was always at least as good as the original classification performance).

Dataset	Model	Accuracy	Precision
Obfuscated Text	BERT	0.494	0.494
	OCR + BERT	**0.595**	0.559
	Vision-enhanced BERT (ours)	0.572	**0.598**
Original Text	BERT	0.852	0.824

Table 1: Comparison of model accuracy and precision on two datasets

Our model improves upon the BERT baseline by 7.8% on accuracy and 10.4% on precision. We underperform the OCR + BERT baseline on accuracy by 2.3% but we beat it on precision by 3.9%. Overall, we achieved our goal of recovering some performance using a vision model to enhance our BERT classifier. Moreover, the fact that we were able to achieve roughly on-par performance with a

professionally developed OCR tool suggests that our approach is promising. With further exploration in text representation in rendering and data augmentation for further alignment between the ViT and BERT embeddings, we can likely do even better. However, we are still far from the BERT's original performance on the original text, which suggests that there is significant headroom for future improvements.

6 Analysis

One crucial area for our method is the quality of embeddings. While we were able to steer our ViT model towards some level of alignment with the BERT embeddings, it is still not quite there. Below are some visualizations of the embeddings for different experiments we ran.

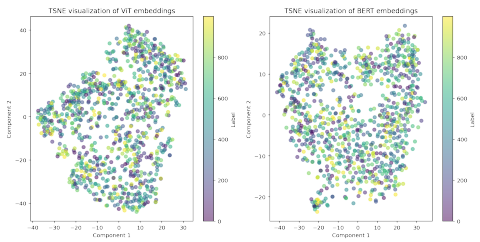

Figure 5: Pretrained ViT

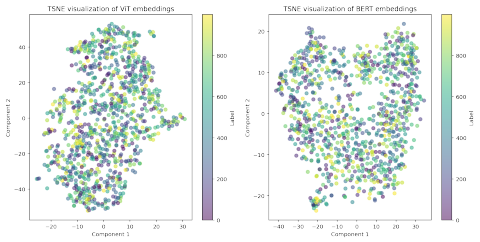

Figure 6: Finetuned ViT with 128-dim projection space

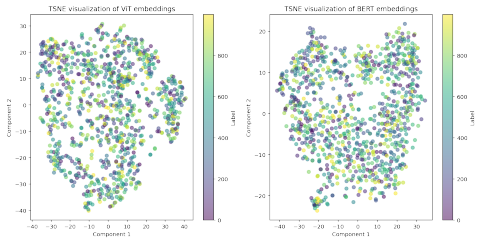

Figure 7: Finetuned ViT with 512-dim projection space, trained for 5 epochs

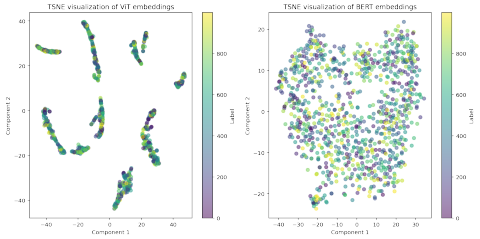

Figure 8: Finetuned ViT with 512-dim projection space, trained for 30 epochs

Figure 9: Comparison of various ViT models against BERT embeddings. The left figures show a dimensionally reduced representations of all of the ViT embeddings for the validation set. The right figures show dimensionally reduced representations of all the BERT embeddings for the validation set. We aspire to have them be the same.

Confusingly, we see that while Figure 8 arguably looks furthest from the associated BERT embeddings we are striving towards, it has the best performance. It is also possible that while the other three representations occupy more volume like the BERT embeddings, they may not actually match in terms of individual embeddings. Overall, the ViT embeddings are not very much aligned with the BERT embeddings at all, which shows us that there is a lot of room for improvement. Theoretically, if we are able to achieve more accurate alignment, we should see a great boost in performance.

7 Conclusion

We introduce a new vision-enhanced BERT model for obfuscated text abuse detection. We demonstrated significant limitations of a vanilla BERT model when faced with obfuscated text and presented a new paradigm. Our model, a Vision-enhanced BERT model that uses contrastive learning to align ViT image embeddings with text embeddings from a fixed BERT model, shows a 7.8% improvement in accuracy and 10.4% improvement in precision over the original model. These results were on par with Python's `pytesseract` OCR tool, which suggests that our approach is a promising one.

In the future, given more exploration in improving the alignment between ViT and BERT embeddings, there is much headroom in further enhancing a BERT model's performance on obfuscated text using

vision. Some avenues of future work might include using various data augmentation techniques to transform the renderings of text so that the ViT model becomes more robust against different kinds of obfuscations (perhaps scaling and warping could be useful after all, as long as the ViT model is able to learn that they are all the same underlying text). It could be useful to explore the literature on how OCR models are trained and adapt those techniques to the finetuning of the ViT model. It would also be interesting to evaluate a Vision-enhanced BERT model on ASCII art obfuscations.

8 Ethics Statement

Suppose we are to deploy such a classifier into production for a social media platform. One ethical challenge could be potential over-enforcement or under-enforcement due to misclassification or choice of operating point. Over-enforcement might look like excessive censorship against users and under-enforcement may leave potentially harmful or violence-inciting content on the platform. As the engineers, we would have to be careful in the development process to mitigate as many false positives and false negatives as possible. Additionally, if we use a classifier that outputs a score and we need to choose a threshold above which a piece of text is considered harmful, we would have to carefully consider what operating point that is. We would have to strike a fair balance between precision and recall to enforce at just the right level to keep users content and to prevent abuse. One way to approach this might be to measure the exact volume of comments that are false positives and false negatives and make a qualitative judgment based on that study. For instance, it is likely that almost all comments on the platform are good comments and very few are bad. In this case, it might be allowable to have a higher false positive rate because the proportion of false positives relative to the entire pool may not be that high.

Another more specific ethical challenge could be evaluating fairness on various textual terms relating to protected groups like those based on gender, sexuality, race, etc. For instance, we want to be fair when classifying comments that contain the words "man" vs. "woman", or various LGBT terms like "gay" or "lesbian" compared with comments that do not contain those words. To address this issue, we could sample a representative subset of the daily comment traffic coming into the model and run an analysis on various categories of protected groups to determine whether or not there is an imbalance. If there is, we could address the problem by augmenting the training data to include more or less comments containing such a term that is over- or under-enforced by the current model.

References

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale.

Devansh Mody, YiDong Huang, and Thiago Eustaquio Alves de Oliveira. 2023. A curated dataset for hate speech detection on social media text. *Data in Brief*, 46:108832.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.