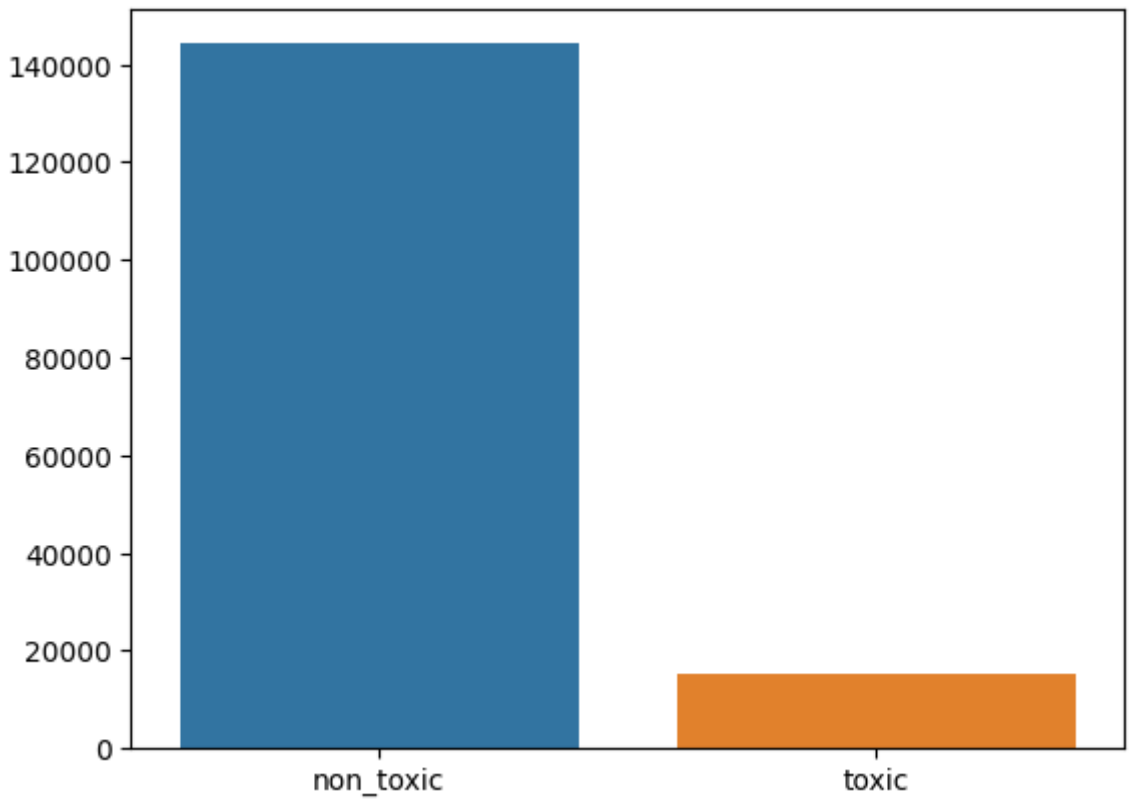
# Team members:

- Nikita Sergeev
- Gia Trong Nguyen

# Current progress:

## Researching phase

- In this phase of the project, we started to investigate an adversarial part of the toxic comments classification task. First of all, we start to search for the papers on the topic of adversarial attacks of the BERT model. And the most interesting work we found on the subject was this article. In this paper authors propose TextFooler model for adversarial attacks on the BERT model for text classification. Then we found some implementations of this model here and started to study them.
- And the first idea we got from the article and the implementation is that the multilabel classification task is too hard for the adversarial attack. So we decided to simplify the task a bit and move to binary classification of the toxic comment.
- For this purpose, we modified the training phase of the model to preprocess the data in a different way. Also in this phase we decided to use less data as we were simplifying the task. After comparing the classes of the data we got next results




So the initial data is also unbalanced. This can affect the result of the training because we can divide the data into training and validation sets in an unfortunate way, so we got 5000 toxic and non-toxic comments for the training set and 500 for the validation set. And with this amount of data after the training phase we still got 92% ROC AUC.

[20]:

| | id | comment_text | toxic | non_toxic |
|---|---|---|---|---|
| **0** | 0000997932d777bf | Explanation\nWhy the edits made under my usern... | 0 | 1 |
| **1** | 000103f0d9cfb60f | D'aww! He matches this background colour I'm s... | 0 | 1 |
| **2** | 000113f07ec002fd | Hey man, I'm really not trying to edit war. It... | 0 | 1 |
| **3** | 0001b41b1c6bb37e | "\nMore\nI can't make any real suggestions on ... | 0 | 1 |
| **4** | 0001d958c54c6e35 | You, sir, are my hero. Any chance you remember... | 0 | 1 |
| **5** | 00025465d4725e87 | "\n\nCongratulations from me as well, use the ... | 0 | 1 |
| **6** | 0002bcb3da6cb337 | COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK | 1 | 0 |
| **7** | 00031b1e95af7921 | Your vandalism to the Matt Shirvington article... | 0 | 1 |
| **8** | 00037261f536c51d | Sorry if the word 'nonsense' was offensive to ... | 0 | 1 |
| **9** | 00040093b2687caa | alignment on this subject and which are contra... | 0 | 1 |

```
EVALUATING at iteration - 300
 96%|█████████| 300/313 [04:31<00:39,  3.02s/it]
ROC_AUC for labels:
 * toxic - 0.923

EVALUATE LOSS -  0.19681932032108307
100%|██████████| 313/313 [04:39<00:00,  1.12it/s]
Train loss 0.43380900596182187


EVALUATING


ROC_AUC for labels:
 * toxic - 0.929

EVALUATE LOSS -  0.20413735508918762
```

Taking less data from the dataset will also help us in the future. For example, while researching the topic of adversarial attacks, we found that generating adversarial samples and making the initial model more robust is a computationally expensive task, so we simply wouldn't have enough resources for this task on the full dataset of toxic comments. The next step for us was to decide how to implement adversarial example generation for our model. We tried to implement the algorithm from scratch, but didn't succeed. And then we decided to generate these examples using an already implemented solution - the `textattack` library, and started to study it's documentation and experiment with the examples from this lib.

## Implementing phase:

After being introduced to the basics of the library, we start to implement adverarial samples generator for our specific model and dataset. For this purpose we firstly implement fasttext model wrapper that looks this way:

```
class MyModelWrapper(ModelWrapper):
        def __init__(self, model, tokenizer):
                self.model = model
                self.tokenizer = tokenizer
        def __call__(self, text_input_list):
                with torch.no_grad():
                        x = self.row_to_tensor(text_input_list).to(device)
                        mask = (x != 0).float()
                        outputs = model(x, attention_mask=mask)
                        final_preds = torch.flip(outputs, dims=(1,))
```

```
                        return final_preds
        def row_to_tensor(self, row):
                tensor = []
                for i in row:
                        tokens = tokenizer.encode(i, add_special_tokens=True)
                        tokens = tokens[:119] + [tokens[-1]] if len(tokens) > 120
else tokens
                        tensor.append(torch.LongTensor(tokens))
                tensor = pad_sequence(tensor, batch_first=True, padding_value=0)
                x = torch.LongTensor(tensor)
                return x
```

Then we tried to generate 100 adversarial samples. We attacked the toxic samples. Goal is to perturbe toxic samples enough that the model classifies them as Non-toxic.
After the 100 atacks on the toxic samples, we got the next results:

```
+--------------------------------+--------+
| Attack Results                 |        |
+--------------------------------+--------+
| Number of successful attacks:  | 73     |
| Number of failed attacks:      | 23     |
| Number of skipped attacks:     | 4      |
| Original accuracy:             | 96.0%  |
| Accuracy under attack:         | 23.0%  |
| Attack success rate:           | 76.04% |
| Average perturbed word %:      | 22.44% |
| Average num. words per input:  | 39.26  |
| Avg num queries:               | 316.01 |
+--------------------------------+--------+
```

And here is an example of text modification for adversarial samples:

| | original_text | perturbed_text |
|---|---|---|
| **95** | When you're an admin, you should try to learn the policy of Wikipedia. Too hard for you! LOLOLOLOL. So I'm a good editor contributing to Wikipedia and you, Swarm, are a total idiot who can't learn the simple policies of WP. LOLOLOLOLOL. Yes, you're a laughable fool! | When you're an administering, you must intends to acquainted the strategy of Wiki. Likewise intensely for you! LOLOLOLOL. Pat I'm a adequate written assistance to Wiki and you, Swarm, are a ensemble fool who can't obtaining the purely statesman of WP. LOLOLOLOLOL. Um, you're a foolish fool! |
| **96** | Tripolis MassacreHecrtoian. Your sick fanatic mind should be deleted from wikipedia. Tripolis massacre is a fact. Soon I'll put in here with GREEK footnotes. Thus, you'll accept it. Are you ashamed of it, or are you just a standart nationalist which, by nature, must oversee some facts. Here in Istanbul there monuments which remind the September 6-7 pogrom. Are there any plates/monuments which remind the turkish population????Few days ago greek historian Prof. Veremis also accepted the Tripolis masscre. Dirty fashist. You can tespace from truth. | Tripolis MassacreHecrtoian. Votre pathological dogmatic mind should be deleted from wikipedia. Tripolis massacre is a fact. Soon I'll put in here with GREEK footnotes. Thus, you'll accept it. Are you ashamed of it, or are you just a standart nationalist which, by nature, must oversee some facts. Here in Istanbul there monuments which remind the September 6-7 pogrom. Are there any plates/monuments which remind the turkish population????Few days ago aegean historian Prof. Veremis also accepted the Tripolis masscre. Dirty fashist. You can tespace from truth. |
| **97** | Oh For CHRISSAKES Deconstructhis, will you stop this bullshit once and for all? Leave me edits ALONE PLEASE... Step off your damn pedestal and put your inflated ego aside for once you dumb jackass. | Oh For GAWD Deconstructhis, dedication you halted this hooey once and for all? Authorization me edits EXCLUSIVE INVITATION... Milestones off your christ pedestal and introduces your overestimated subjectivity aside for once you ironic asinine. |

So after the attack, we reduce the accuracy of the model from 96% to 23% for the given 100 toxic samples. At this stage of the project's development, we think this is a pretty good result.

## Each member contribution:

- Nikita - researching about BERT robustness, retraining model for binary classification and changing the data, report writing.
- Gia Trong - reseaching about implementation of adverarial attcks libs, implementing attack on the current model and logging attacks.

## Future work:

Over the next 3 weeks, we plan to develop adversarial part of this project. For now, we have the following ideas:

1. Experement with the adverarial samples generator, try to udenstand it's design and logic and try to get results, simmilar to lib built-in function s with the solution implemented from scratch.
2. Train the model on adversarial examples.
3. Evaluate the model's robustness.
4. Fine-tune the model, using generated adversarial examples.
   Nonetheless, this project's development plan is subject to change. Throughout the research process, the team may discover other experiments that they find intriguing and would like to incorporate into the model. While these experiments may not significantly impact the model's performance, they present a valuable opportunity for the team to hone their machine learning skills.