In [ ]:

In [ ]:

In [ ]:

In [2]:
```
pip install sentencepiece
```

```
Collecting sentencepiece
  Downloading sentencepiece-0.2.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014
_x86_64.whl.metadata (7.7 kB)
Downloading sentencepiece-0.2.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x
86_64.whl (1.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.3/1.3 MB 18.5 MB/s eta 0:00:00
Installing collected packages: sentencepiece
Successfully installed sentencepiece-0.2.0
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
pip install torch
```

```
Collecting torch
  Downloading torch-2.6.0-cp312-cp312-manylinux1_x86_64.whl.metadata (28 kB)
Collecting filelock (from torch)
  Downloading filelock-3.18.0-py3-none-any.whl.metadata (2.9 kB)
Requirement already satisfied: typing-extensions>=4.10.0 in /opt/conda/lib/python
3.12/site-packages (from torch) (4.12.2)
Requirement already satisfied: networkx in /opt/conda/lib/python3.12/site-package
s (from torch) (3.4.2)
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.12/site-packages
(from torch) (3.1.5)
Requirement already satisfied: fsspec in /opt/conda/lib/python3.12/site-packages
(from torch) (2025.2.0)
Collecting nvidia-cuda-nvrtc-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.m
etadata (1.5 kB)
Collecting nvidia-cuda-runtime-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_64.wh
l.metadata (1.5 kB)
Collecting nvidia-cuda-cupti-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.m
etadata (1.6 kB)
Collecting nvidia-cudnn-cu12==9.1.0.70 (from torch)
  Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl.metada
ta (1.6 kB)
Collecting nvidia-cublas-cu12==12.4.5.8 (from torch)
  Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl.metad
ata (1.5 kB)
Collecting nvidia-cufft-cu12==11.2.1.3 (from torch)
  Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl.metada
ta (1.5 kB)
Collecting nvidia-curand-cu12==10.3.5.147 (from torch)
  Downloading nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.whl.met
adata (1.5 kB)
Collecting nvidia-cusolver-cu12==11.6.1.9 (from torch)
  Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.whl.met
adata (1.6 kB)
Collecting nvidia-cusparse-cu12==12.3.1.170 (from torch)
  Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.whl.m
etadata (1.6 kB)
Collecting nvidia-cusparselt-cu12==0.6.2 (from torch)
  Downloading nvidia_cusparselt_cu12-0.6.2-py3-none-manylinux2014_x86_64.whl.meta
data (6.8 kB)
Collecting nvidia-nccl-cu12==2.21.5 (from torch)
  Downloading nvidia_nccl_cu12-2.21.5-py3-none-manylinux2014_x86_64.whl.metadata
(1.8 kB)
Collecting nvidia-nvtx-cu12==12.4.127 (from torch)
  Downloading nvidia_nvtx_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadat
a (1.7 kB)
Collecting nvidia-nvjitlink-cu12==12.4.127 (from torch)
  Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.me
tadata (1.5 kB)
Collecting triton==3.2.0 (from torch)
  Downloading triton-3.2.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl.metadata (1.4 kB)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.12/site-packa
ges (from torch) (75.8.0)
Collecting sympy==1.13.1 (from torch)
  Downloading sympy-1.13.1-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /opt/conda/lib/python3.12/si
te-packages (from sympy==1.13.1->torch) (1.3.0)
```

```
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.12/site-
packages (from jinja2->torch) (3.0.2)
Downloading torch-2.6.0-cp312-cp312-manylinux1_x86_64.whl (766.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 766.6/766.6 MB 26.8 MB/s eta 0:00:000
0:0100:01
Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl (363.4
MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 43.8 MB/s eta 0:00:000
0:0100:01
Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (1
3.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 45.5 MB/s eta 0:00:0000:
01
Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (2
4.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 52.8 MB/s eta 0:00:0000:
0100:01
Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl
(883 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 58.9 MB/s eta 0:00:00
Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl (664.8 M
B)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 43.0 MB/s eta 0:00:000
0:0100:01
Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl (211.5 M
B)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 68.1 MB/s eta 0:00:000
0:0100:01
Downloading nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.whl (56.3
MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 48.8 MB/s eta 0:00:0000:
0100:01
Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.whl (127.
9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 59.2 MB/s eta 0:00:000
0:0100:01
Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.whl (20
7.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 58.5 MB/s eta 0:00:000
0:0100:01
Downloading nvidia_cusparselt_cu12-0.6.2-py3-none-manylinux2014_x86_64.whl (150.1
MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 150.1/150.1 MB 66.4 MB/s eta 0:00:000
0:0100:01
Downloading nvidia_nccl_cu12-2.21.5-py3-none-manylinux2014_x86_64.whl (188.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 188.7/188.7 MB 56.5 MB/s eta 0:00:000
0:0100:01
Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (21.
1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 56.5 MB/s eta 0:00:00a
0:00:01
Downloading nvidia_nvtx_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (99 kB)
Downloading sympy-1.13.1-py3-none-any.whl (6.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.2/6.2 MB 59.7 MB/s eta 0:00:00
Downloading triton-3.2.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.w
hl (253.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 253.2/253.2 MB 56.3 MB/s eta 0:00:000
0:0100:01
Downloading filelock-3.18.0-py3-none-any.whl (16 kB)
Installing collected packages: triton, nvidia-cusparselt-cu12, sympy, nvidia-nvtx
```

```
-cu12, nvidia-nvjitlink-cu12, nvidia-nccl-cu12, nvidia-curand-cu12, nvidia-cufft-
cu12, nvidia-cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12, nvidia-cuda-cupti-cu12, n
vidia-cublas-cu12, filelock, nvidia-cusparse-cu12, nvidia-cudnn-cu12, nvidia-cuso
lver-cu12, torch
  Attempting uninstall: sympy
    Found existing installation: sympy 1.13.3
    Uninstalling sympy-1.13.3:
      Successfully uninstalled sympy-1.13.3
Successfully installed filelock-3.18.0 nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cu
pti-cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.127 nvidia-cuda-runtime-cu12-12.4.1
27 nvidia-cudnn-cu12-9.1.0.70 nvidia-cufft-cu12-11.2.1.3 nvidia-curand-cu12-10.3.
5.147 nvidia-cusolver-cu12-11.6.1.9 nvidia-cusparse-cu12-12.3.1.170 nvidia-cuspar
selt-cu12-0.6.2 nvidia-nccl-cu12-2.21.5 nvidia-nvjitlink-cu12-12.4.127 nvidia-nvt
x-cu12-12.4.127 sympy-1.13.1 torch-2.6.0 triton-3.2.0
Note: you may need to restart the kernel to use updated packages.
```

In [4]:
```
pip install transformers
```

```
Collecting transformers
  Downloading transformers-4.50.3-py3-none-any.whl.metadata (39 kB)
Requirement already satisfied: filelock in /opt/conda/lib/python3.12/site-package
s (from transformers) (3.18.0)
Collecting huggingface-hub<1.0,>=0.26.0 (from transformers)
  Downloading huggingface_hub-0.30.1-py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: numpy>=1.17 in /opt/conda/lib/python3.12/site-pack
ages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.12/site-
packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /opt/conda/lib/python3.12/site-pack
ages (from transformers) (6.0.2)
Collecting regex!=2019.12.17 (from transformers)
  Downloading regex-2024.11.6-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86
_64.whl.metadata (40 kB)
Requirement already satisfied: requests in /opt/conda/lib/python3.12/site-package
s (from transformers) (2.32.3)
Collecting tokenizers<0.22,>=0.21 (from transformers)
  Downloading tokenizers-0.21.1-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86
_64.whl.metadata (6.8 kB)
Collecting safetensors>=0.4.3 (from transformers)
  Downloading safetensors-0.5.3-cp38-abi3-manylinux_2_17_x86_64.manylinux2014_x86
_64.whl.metadata (3.8 kB)
Requirement already satisfied: tqdm>=4.27 in /opt/conda/lib/python3.12/site-packa
ges (from transformers) (4.67.1)
Requirement already satisfied: fsspec>=2023.5.0 in /opt/conda/lib/python3.12/site
-packages (from huggingface-hub<1.0,>=0.26.0->transformers) (2025.2.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /opt/conda/lib/pytho
n3.12/site-packages (from huggingface-hub<1.0,>=0.26.0->transformers) (4.12.2)
Requirement already satisfied: charset_normalizer<4,>=2 in /opt/conda/lib/python
3.12/site-packages (from requests->transformers) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.12/site-pac
kages (from requests->transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/conda/lib/python3.12/si
te-packages (from requests->transformers) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.12/si
te-packages (from requests->transformers) (2024.12.14)
Downloading transformers-4.50.3-py3-none-any.whl (10.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.2/10.2 MB 14.6 MB/s eta 0:00:0000:
01
Downloading huggingface_hub-0.30.1-py3-none-any.whl (481 kB)
Downloading regex-2024.11.6-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (796 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 796.9/796.9 kB 1.4 MB/s eta 0:00:00:0
0:01
Downloading safetensors-0.5.3-cp38-abi3-manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (471 kB)
Downloading tokenizers-0.21.1-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (3.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.0/3.0 MB 6.3 MB/s eta 0:00:00ta 0:0
0:01
Installing collected packages: safetensors, regex, huggingface-hub, tokenizers, t
ransformers
Successfully installed huggingface-hub-0.30.1 regex-2024.11.6 safetensors-0.5.3 t
okenizers-0.21.1 transformers-4.50.3
Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```python
import torch
import torch.nn as nn
from torch.utils.data import DataLoader, Dataset, random_split
```

```
from transformers import MT5Tokenizer, MT5ForSequenceClassification
import pandas as pd
from tqdm import tqdm
import os

class TextDataset(Dataset):
    def __init__(self, texts, labels, tokenizer, max_length=128):
        self.texts = texts
        self.labels = labels
        self.tokenizer = tokenizer
        self.max_length = max_length

    def __len__(self):
        return len(self.texts)

    def __getitem__(self, idx):
        encoding = self.tokenizer(
            self.texts[idx],
            truncation=True,
            padding='max_length',
            max_length=self.max_length,
            return_tensors="pt"
        )
        return {
            'input_ids': encoding['input_ids'].squeeze(0),
            'attention_mask': encoding['attention_mask'].squeeze(0),
            'labels': torch.tensor(self.labels[idx], dtype=torch.long)
        }

train_df = pd.read_csv('toxic_eng/train.csv')
test_df = pd.read_csv('toxic_eng/test.csv')

train_texts = train_df['comment_text'].tolist()
train_labels = train_df['toxic'].tolist()
test_texts = test_df['comment_text'].tolist()
test_labels = test_df['toxic'].tolist()

tokenizer = MT5Tokenizer.from_pretrained("google/mt5-base")

MAX_LEN = 128
full_train_dataset = TextDataset(train_texts, train_labels, tokenizer, max_lengt
test_dataset = TextDataset(test_texts, test_labels, tokenizer, max_length=MAX_LE

train_size = int(0.9 * len(full_train_dataset))
val_size = len(full_train_dataset) - train_size
train_dataset, val_dataset = random_split(full_train_dataset, [train_size, val_s

train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=32)
test_loader = DataLoader(test_dataset, batch_size=32)

def save_checkpoint(model, optimizer, epoch, checkpoint_dir, model_name, best_va
    os.makedirs(checkpoint_dir, exist_ok=True)
    checkpoint_path = os.path.join(checkpoint_dir, f"{model_name}_epoch{epoch+1}
    torch.save({
        'epoch': epoch + 1,
        'model_state_dict': model.state_dict(),
        'optimizer_state_dict': optimizer.state_dict(),
        'best_val_loss': best_val_loss
    }, checkpoint_path)
```

```
    print(f"Checkpoint saved at: {checkpoint_path}")

def compute_val_loss(model, val_loader, criterion, device):
    model.eval()
    total_loss = 0
    with torch.no_grad():
        for batch in val_loader:
            input_ids = batch['input_ids'].to(device)
            attention_mask = batch['attention_mask'].to(device)
            labels = batch['labels'].to(device)
            outputs = model(input_ids=input_ids, attention_mask=attention_mask)
            loss = criterion(outputs.logits, labels)
            total_loss += loss.item()
    return total_loss / len(val_loader)

def load_checkpoint(model, optimizer, checkpoint_dir, model_name):
    checkpoint_path = os.path.join(checkpoint_dir, f"{model_name}_best.pt")
    if os.path.exists(checkpoint_path):
        checkpoint = torch.load(checkpoint_path)
        model.load_state_dict(checkpoint['model_state_dict'])
        optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
        epoch = checkpoint['epoch']
        best_val_loss = checkpoint['best_val_loss']
        print(f"Resuming training from epoch {epoch + 1} with best validation lo
        return epoch, best_val_loss
    else:
        print("No checkpoint found. Starting training from scratch.")
        return 0, float('inf')

def train_with_early_stopping(model, train_loader, val_loader, criterion, optimi
    best_val_loss = float('inf')
    patience_counter = 0

    epoch, best_val_loss = load_checkpoint(model, optimizer, checkpoint_dir, mod

    for epoch in range(epoch, num_epochs):
        model.train()
        total_loss = 0
        correct = 0
        total = 0
        loop = tqdm(train_loader, desc=f"Epoch {epoch+1}/{num_epochs}")

        for batch in loop:
            input_ids = batch['input_ids'].to(device)
            attention_mask = batch['attention_mask'].to(device)
            labels = batch['labels'].to(device)

            outputs = model(input_ids=input_ids, attention_mask=attention_mask)
            logits = outputs.logits
            loss = criterion(logits, labels)

            optimizer.zero_grad()
            loss.backward()
            optimizer.step()

            total_loss += loss.item()
            _, predicted = torch.max(logits, dim=1)
            correct += (predicted == labels).sum().item()
            total += labels.size(0)
```

```
            loop.set_postfix(loss=loss.item())

        train_accuracy = correct / total
        val_loss = compute_val_loss(model, val_loader, criterion, device)
        print(f"Epoch {epoch+1}, Train Loss: {total_loss:.4f}, Train Acc: {train

        save_checkpoint(model, optimizer, epoch, checkpoint_dir, model_name, bes

        # Save best model
        if val_loss < best_val_loss:
            best_val_loss = val_loss
            patience_counter = 0
            best_path = os.path.join(checkpoint_dir, f"{model_name}_best.pt")
            torch.save({
                'epoch': epoch + 1,
                'model_state_dict': model.state_dict(),
                'optimizer_state_dict': optimizer.state_dict(),
                'best_val_loss': best_val_loss
            }, best_path)
            print(f"Best model saved at: {best_path}")
        else:
            patience_counter += 1
            print(f"Early stopping patience: {patience_counter}/{patience}")
            if patience_counter >= patience:
                print("Early stopping triggered.")
                break


def evaluate_mt5_model(model, test_loader, device):
    model.eval()
    correct = 0
    total = 0
    test_loader_tqdm = tqdm(test_loader, desc="Evaluating", leave=False)
    with torch.no_grad():
        for batch in test_loader_tqdm:
            input_ids = batch['input_ids'].to(device)
            attention_mask = batch['attention_mask'].to(device)
            labels = batch['labels'].to(device)
            outputs = model(input_ids, attention_mask)
            logits = outputs.logits
            _, preds = torch.max(logits, 1)
            correct += (preds == labels).sum().item()
            total += labels.size(0)
    accuracy = correct / total
    print(f"Test Accuracy: {accuracy:.4f}")

# Init
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = MT5ForSequenceClassification.from_pretrained("google/mt5-base", num_labe
optimizer = torch.optim.Adam(model.parameters(), lr=1e-5)
criterion = nn.CrossEntropyLoss()
```

```
The tokenizer class you load from this checkpoint is not the same type as the cla
ss this function is called from. It may result in unexpected tokenization.
The tokenizer class you load from this checkpoint is 'T5Tokenizer'.
The class this function is called from is 'MT5Tokenizer'.
Some weights of MT5ForSequenceClassification were not initialized from the model
checkpoint at google/mt5-base and are newly initialized: ['classification_head.de
nse.bias', 'classification_head.dense.weight', 'classification_head.out_proj.bia
s', 'classification_head.out_proj.weight']
You should probably TRAIN this model on a down-stream task to be able to use it f
or predictions and inference.
```

In [ ]:
```
# Train on full dataset
train_with_early_stopping(
    model, train_loader, val_loader,
    criterion, optimizer,
    device, num_epochs=10,
    checkpoint_dir="checkpoints_full",
    model_name="mt5_toxicity_full",
    patience=3
)

# Evaluate
evaluate_mt5_model(model, test_loader, device)
```

In [6]:
```
# Load best checkpoint for evaluation
def load_best_model(model_class, checkpoint_dir, model_name, device):
    checkpoint_path = os.path.join(checkpoint_dir, f"{model_name}_best.pt")
    if os.path.exists(checkpoint_path):
        print(f"Loading best model from {checkpoint_path}")
        checkpoint = torch.load(checkpoint_path, map_location=device)
        model = model_class.from_pretrained("google/mt5-base", num_labels=2)
        model.load_state_dict(checkpoint['model_state_dict'])
        model.to(device)
        return model
    else:
        raise FileNotFoundError(f"Best checkpoint not found at {checkpoint_path}

best_model = load_best_model(MT5ForSequenceClassification, "checkpoints_full", "
evaluate_mt5_model(best_model, test_loader, device)
```

```
Loading best model from checkpoints_full/mt5_toxicity_full_best.pt
Some weights of MT5ForSequenceClassification were not initialized from the model
checkpoint at google/mt5-base and are newly initialized: ['classification_head.de
nse.bias', 'classification_head.dense.weight', 'classification_head.out_proj.bia
s', 'classification_head.out_proj.weight']
You should probably TRAIN this model on a down-stream task to be able to use it f
or predictions and inference.

Test Accuracy: 0.9405
```

In [8]:
```
# Load best checkpoint for evaluation
def load_best_model(model_class, checkpoint_dir, model_name, device):
    checkpoint_path = os.path.join(checkpoint_dir, f"{model_name}.pt")
    if os.path.exists(checkpoint_path):
        print(f"Loading best model from {checkpoint_path}")
        checkpoint = torch.load(checkpoint_path, map_location=device)
        model = model_class.from_pretrained("google/mt5-base", num_labels=2)
        model.load_state_dict(checkpoint['model_state_dict'])
        model.to(device)
```

```
        return model
    else:
        raise FileNotFoundError(f"Best checkpoint not found at {checkpoint_path}

# Load best model and evaluate it on test set
best_model = load_best_model(MT5ForSequenceClassification, "checkpoints_full", "
evaluate_mt5_model(best_model, test_loader, device)
```

```
Loading best model from checkpoints_full/mt5_toxicity_full_epoch10.pt
Some weights of MT5ForSequenceClassification were not initialized from the model
checkpoint at google/mt5-base and are newly initialized: ['classification_head.de
nse.bias', 'classification_head.dense.weight', 'classification_head.out_proj.bia
s', 'classification_head.out_proj.weight']
You should probably TRAIN this model on a down-stream task to be able to use it f
or predictions and inference.

Test Accuracy: 0.9423
```

In [ ]:
```
import os

checkpoint_dir = "checkpoints"
files = os.listdir(checkpoint_dir)
print("Obsah priečinka checkpoints:")
for file in files:
    print(f"- {file}")
```

In [9]:
```
import torch
from transformers import MT5ForSequenceClassification
from sklearn.metrics import classification_report, confusion_matrix
from tqdm import tqdm
import os
# Load full dataset
train_df = pd.read_csv('toxic_eng/train.csv')
test_df = pd.read_csv('toxic_eng/test.csv')

train_texts = train_df['comment_text'].tolist()
train_labels = train_df['toxic'].tolist()
test_texts = test_df['comment_text'].tolist()
test_labels = test_df['toxic'].tolist()

# Tokenizer
tokenizer = MT5Tokenizer.from_pretrained("google/mt5-base")
# Načítanie checkpointu
def load_best_model(model_class, checkpoint_dir, model_name, device):
    checkpoint_path = os.path.join(checkpoint_dir, f"{model_name}.pt")
    if os.path.exists(checkpoint_path):
        print(f"Loading best model from {checkpoint_path}")
        checkpoint = torch.load(checkpoint_path, map_location=device)
        model = model_class.from_pretrained("google/mt5-base", num_labels=2)
        model.load_state_dict(checkpoint['model_state_dict'])
        model.to(device)
        return model
    else:
        raise FileNotFoundError(f"Best checkpoint not found at {checkpoint_path}

# Vyhodnotenie
def evaluate_mt5_model(model, test_loader, device):
    model.eval()
    all_preds = []
```

```
    all_labels = []

    with torch.no_grad():
        for batch in tqdm(test_loader, desc="Vyhodnocovanie"):
            input_ids = batch['input_ids'].to(device)
            attention_mask = batch['attention_mask'].to(device)
            labels = batch['labels'].to(device)

            outputs = model(input_ids=input_ids, attention_mask=attention_mask)
            logits = outputs.logits
            preds = torch.argmax(logits, dim=1)

            all_preds.extend(preds.cpu().numpy())
            all_labels.extend(labels.cpu().numpy())

    print(f"\nPresnosť na testovacích dátach: {torch.sum(torch.tensor(all_preds)
    print("\nKlasifikačná správa:")
    print(classification_report(all_labels, all_preds, digits=4))
    print("\nConfusion Matrix:")
    print(confusion_matrix(all_labels, all_preds))


# === Inicializácia ===
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
best_model = load_best_model(MT5ForSequenceClassification, "checkpoints_full", "

# === Vyhodnotenie ===
evaluate_mt5_model(best_model, test_loader, device)
```

```
The tokenizer class you load from this checkpoint is not the same type as the cla
ss this function is called from. It may result in unexpected tokenization.
The tokenizer class you load from this checkpoint is 'T5Tokenizer'.
The class this function is called from is 'MT5Tokenizer'.
Loading best model from checkpoints_full/mt5_toxicity_full_epoch10.pt
Some weights of MT5ForSequenceClassification were not initialized from the model
checkpoint at google/mt5-base and are newly initialized: ['classification_head.de
nse.bias', 'classification_head.dense.weight', 'classification_head.out_proj.bia
s', 'classification_head.out_proj.weight']
You should probably TRAIN this model on a down-stream task to be able to use it f
or predictions and inference.
Vyhodnocovanie: 100%|██████████| 625/625 [17:05<00:00,  1.64s/it]
Presnosť na testovacích dátach: 0.9423

Klasifikačná správa:
              precision    recall  f1-score   support

           0     0.9623    0.9206    0.9410     10000
           1     0.9239    0.9639    0.9435     10000

    accuracy                         0.9423     20000
   macro avg     0.9431    0.9423    0.9422     20000
weighted avg     0.9431    0.9423    0.9422     20000


Confusion Matrix:
[[9206  794]
 [ 361 9639]]
```

In [ ]: