

ДОДАТОК А

Код перефразування повідомлень

```
from transformers import T5Tokenizer, T5ForConditionalGeneration

model_name = "t5-base"
tokenizer = T5Tokenizer.from_pretrained(model_name)
model = T5ForConditionalGeneration.from_pretrained(model_name)

def rephrase_text(text):
    input_text = "paraphrase: " + text + " </s>"
    input_ids = tokenizer.encode(input_text, return_tensors="pt")

    outputs = model.generate(
        input_ids,
        max_length=256,
        num_return_sequences=1,
        num_beams=5,
        early_stopping=True
    )

    rephrased_text = tokenizer.decode(outputs[0], skip_special_tokens=True)
    return rephrased_text

original_text = """
Hey there, Hope you're doing well! I just wanted to share some exciting news about our
upcoming project. We've made a lot of progress over the past few months, and I thought you'd
like to hear about it. First off, our team has wrapped up the initial phase of the project,
which involved a lot of research and data collection. We've gained some really valuable insights
that will help guide our next steps. Now, we're moving into the development phase where we'll
start implementing all the strategies we've come up with. Also, I'd love for you to join a
meeting next Monday at 10 AM to chat about the project's progress and get your feedback. Your
input is super important to us, and we think it will be instrumental in making this project a
success. Lastly, I just want to say thanks for your continued support and dedication. Together,
we've got the potential to achieve some amazing results, and I'm confident that our combined
efforts will lead to a successful outcome. Looking forward to seeing you at the meeting!"""

rephrased_text = rephrase_text(original_text)
print("Original:", original_text)
print("Rephrased:", rephrased_text)
```

Рисунок А.1 — Перефразування за допомогою моделі T5

```

class BertRephraser:
    def __init__(self, model_name='patrickvonplaten/bert2bert_cnn_dailymail-fp16'):
        self.tokenizer = AutoTokenizer.from_pretrained(model_name)
        self.model = EncoderDecoderModel.from_pretrained(model_name)

    def rephrase(self, text, num_return_sequences=1):
        inputs = self.tokenizer(
            text,
            return_tensors="pt",
            max_length=512,
            truncation=True,
        )
        with torch.no_grad():
            outputs = self.model.generate(
                input_ids=inputs["input_ids"],
                attention_mask=inputs["attention_mask"],
                max_length=512,
                num_return_sequences=num_return_sequences,
                num_beams=5,
                early_stopping=True,
            )

        rephrased_sentences = [
            self.tokenizer.decode(output, skip_special_tokens=True, clean_up_tokenization_spaces=True)
            for output in outputs
        ]
        return rephrased_sentences

```

Рисунок А.2 — Перефразування за допомогою моделі BERT

```

import torch
from transformers import RobertaForMaskedLM, RobertaTokenizer

class TextRephraser:
    def __init__(self, model_name='roberta-base'):
        self.tokenizer = RobertaTokenizer.from_pretrained(model_name)
        self.model = RobertaForMaskedLM.from_pretrained(model_name)

    def rephrase(self, text, mask_token='<mask>'):
        inputs = self.tokenizer(text, return_tensors='pt')
        mask_token_index = torch.where(inputs.input_ids == self.tokenizer.mask_token_id)[1]

        with torch.no_grad():
            logits = self.model(**inputs).logits

        mask_token_logits = logits[0, mask_token_index, :]
        predicted_token_id = torch.argmax(mask_token_logits, axis=-1)
        predicted_token = self.tokenizer.decode(predicted_token_id)

        rephrased_text = text.replace(mask_token, predicted_token)
        return rephrased_text

if __name__ == "__main__":
    rephraser = TextRephraser()
    original_text = "This is a <mask> example of text rephrasing."
    rephrased_text = rephraser.rephrase(original_text)
    print(f"Original: {original_text}")
    print(f"Rephrased: {rephrased_text}")

```

Рисунок А.3 — Перефразування за допомогою моделі RoBERTa

```
import openai

openai.api_key = 'ваш_ключ_API'

def paraphrase_text(text):
    response = openai.Completion.create(
        engine="text-davinci-003",
        prompt=f"Перепаразуйте наступний текст: \"{text}\"",
        max_tokens=100,
        n=1,
        stop=None,
        temperature=0.7,
    )

    paraphrased_text = response.choices[0].text.strip()
    return paraphrased_text

original_text = "Осінь - це пора року, коли листя на деревах змінює колір і починає падати на землю."
paraphrased_text = paraphrase_text(original_text)
print(paraphrased_text)
```

Рисунок А.4 — Перефразування за допомогою моделі GPT

