ДОДАТОК А

Код асинхронного запиту Google Fact-Check з локальним кешем JSON

Лістинг А.1 – Асинхронний запит Google Fact-Check з локальним кешем JSON

```
import aiohttp, asyncio, json, hashlib, os, pathlib, time

CACHE_PATH = pathlib.Path("fact_cache.json")

CACHE      = json.loads(CACHE_PATH.read_text()) if
CACHE_PATH.exists() else {}


BASE_URL   = (

    "https://factchecktools.googleapis.com/"

    "v1alpha1/claims:search"

)


async def _one_request(session, text: str, api_key: str,
lang: str = "en-US"):

    """

    Query Google Fact-Check Tools for the first 120
characters of *text*.

    Result is cached on disk so subsequent calls are free.

    """

    fingerprint =
hashlib.sha1(text[:120].encode()).hexdigest()

    if fingerprint in CACHE:

        return CACHE[fingerprint]


    params = {

        "query": text[:120],

        "languageCode": lang,

        "key": api_key

    }
```

Продовження лістингу А.1

```
    async with session.get(BASE_URL, params=params,
timeout=15) as resp:
        payload = await resp.json()
        CACHE[fingerprint] = payload
        if time.time() % 60 < 1:
            CACHE_PATH.write_text(json.dumps(CACHE))
        return payload



async def factcheck_batch(statements: list[str], api_key:
str) -> list[dict]:
    """
    Fire off ≤ 50 concurrent look-ups (Google quota ≈ 1800
req/min).
    Returns list[JSON] aligned with *statements* order.
    """
    semaphore = asyncio.Semaphore(50)

    async def limited(text):
        async with semaphore:
            async with aiohttp.ClientSession() as sess:
                return await _one_request(sess, text,
api_key)

    tasks = [limited(s) for s in statements]
    return await asyncio.gather(*tasks)
```

ДОДАТОК Б

Скорочена версія коду основної логіки

Лістинг Б.1 – лаконічний, комплексний фрагмент Python, що відображає основну логіку

```
import io, math, tempfile
import numpy as np
from PIL import Image, ImageChops, ImageEnhance
from transformers import AutoTokenizer,
AutoModelForSequenceClassification
import whisper


asr = whisper.load_model("tiny")
def transcribe_ua(video_path: str) -> str:
    """Return Ukrainian transcript (first 20 s for
demo)."""
    out = asr.transcribe(video_path, language="uk",
fp16=False, duration=20)
    return out["text"]


tok =
AutoTokenizer.from_pretrained("MiniLM_finetuned_claims")
clf =
AutoModelForSequenceClassification.from_pretrained("MiniLM
_finetuned_claims")


def claim_score(sentence: str) -> float:
    """Return probability that the statement is false."""
    X = tok(sentence, return_tensors="pt")
    logits = clf(**X).logits.detach().squeeze()
    prob_false = float(logits.softmax(-1)[1])
    return prob_false
```

Продовження лістингу Б.1

```
def ela_score(img: Image.Image, quality: int = 95) ->
float:
    """Return simple ELA tamper likelihood (0 = clean, 1 =
heavy editing)."""
    buf = io.BytesIO()
    img.save(buf, "JPEG", quality=quality)
    ela_img = ImageChops.difference(img, Image.open(buf))
    extrema = ela_img.getextrema()
    max_diff = max(v for _, v in extrema)
    normalized = max_diff / 255.0
    return float(min(1.0, normalized * 3.5))


def fuse(p_text: float, p_vis: float, w_txt: float = 0.6,
w_vis: float = 0.4):
    """Calibrated log-odds fusion of text and vision
probabilities."""
    def logit(p): return math.log(p / (1 - p + 1e-9))
    log_odds = w_txt * logit(p_text) + w_vis *
logit(p_vis)
    return 1 / (1 + math.exp(-log_odds))


if __name__ == "__main__":
    VIDEO = "sample_clip.mp4"
    FRAME = "sample_frame.jpg"


    transcript = transcribe_ua(VIDEO)
    main_claim = transcript.split(".")[0]
    p_text = claim_score(main_claim)


    p_vis = ela_score(Image.open(FRAME))


    p_final = fuse(p_text, p_vis)
    verdict = "FAKE" if p_final > 0.5 else "TRUE"
```

Продовження лістингу Б.1

```
    print(f"Text-prob = {p_text:.2f}  |  Vis-prob =
{p_vis:.2f}")

    print(f"Fused prob = {p_final:.2f}  ➔  {verdict}")
```

ДОДАТОК В

Відомість кваліфікаційної роботи

Позначення	Найменування	Дод. відомості
	Текстові документи	
1.	Пояснювальна записка	78 с.
	Інші документи	
2.	Презентаційні матеріали	__ плакатів

.	.	Прізвище та ініціали.	Підп.	Дата			
Розробив		Болотніков З.В.			Мультимодальна система обробки та аналізу коротких новинних відео	Шифр групи	Код напр./спец.
Перевірив		Чала Л.Е.				ІТШІ-21-2	122
Н.контр.		Малєєва І.А.				ХНУРЕ кафедра ШІ	
Затв.		Золотухін О.В.					