Papagáj CK Košice, 16. -18. 4. 2026

Slovenský mediálny watchdog zaznamenal znepokojujúci trend. V posledných mesiacoch sa na spravodajských portáloch, v parlamentných súhrnoch, ale aj na Wikipédii objavujú texty, ktoré nikto nenapísal. Teda, napísať ich niekto napísal, len to bol podľa všetkého stroj. Veľké jazykové modely (LLMs) už dokážu generovať slovenský text takmer na nerozoznanie od ľudského. A ako sa ukazuje, hodne sa to zneužíva.

Operácia na odhalenie týchto textov dostala krycie meno **Papagáj** -- lebo predsa len, tie stochastické papagáje, a tak...

Vašou úlohou je stať sa analytikom operácie Papagáj. Dostanete dataset slovenských textov -- niektoré napísali ľudia, iné vygenerovali štyri rôzne jazykové modely. Vašou úlohou je:

1. Analyzovať dataset a zistiť jeho štatistické vlastnosti
2. Rozlíšiť ľudské texty od strojových
3. Identifikovať, ktorý model vygeneroval ktorý text
4. Odhaliť aj texty, ktoré boli prepísané tak, aby vyzerali ľudsky

---

Prehľad úlohy

Podúloha	Názov	Body	Popis
1	Prieskum	10	Štatistická analýza testovacieho datasetu
2	Identifikácia	15	Binárna klasifikácia: ľudský vs. strojový text
3	Atribúcia	45	Identifikácia zdrojového modelu pre strojové texty
4	Kontrarozviedka	30	Detekcia parafrázovaných strojových textov

Spoločný vstup: Slovenské texty z parlamentných prejavov, spravodajských článkov a Wikipédie.

Zdroje ľudských textov:

- Slovenský parlament (prepisy z NRSR)
- Slovenské spravodajské články
- Slovenská Wikipédia

Strojové texty boli vygenerované štyrmi modelmi:

- GPT-5.4 (OpenAI)
- Claude Sonnet 4.6 (Anthropic)
- Gemini 3.1 Pro (Google)
- Gemma 4 31B (Google, open-source)

Podúloha 1 -- Prieskum (10 bodov)
Spočítajte nasledujúce štatistiky na **celom testovacom datasete** (súbor test_s2.jsonl). Každá štatistika je jeden riadok vo výstupnom CSV.

Riadok	Štatistika	Popis
0	Percento strojových textov	Zaokrúhlite na 2 desatinné miesta
1	Celkový počet textov	Celé číslo
2	Priemerná dĺžka (slová)	Zaokrúhlite na 1 desatinné miesto
3	Medián dĺžky (slová)	Zaokrúhlite na 1 desatinné miesto
4	Priemerná dĺžka (znaky)	Zaokrúhlite na 1 desatinné miesto
5	Priemerný počet tokenov	Použite tiktoken tokenizér o200k_base. Zaokrúhlite na 1 des. miesto
6	Priemerný počet slov na vetu	Zaokrúhlite na 1 desatinné miesto
7	Priemerná dĺžka slova (znaky)	Zaokrúhlite na 1 desatinné miesto

Riadok	Štatistika	Popis
8	Priemerný type-token ratio	Unikátne slová / celkový počet slov. Zaokrúhlite na 4 des. miesta
9	Freq. najčastejšieho bigramu	Priemer cez texty. Zaokrúhlite na 4 desatinné miesta
10	Freq. najčastejšieho trigramu	Priemer cez texty. Zaokrúhlite na 4 desatinné miesta
11	Priemerný počet interpunkcií	Na text. Zaokrúhlite na 1 desatinné miesto
12	Q1 dĺžky (slová)	25. percentil. Zaokrúhlite na 1 des. miesto
13	Q3 dĺžky (slová)	75. percentil. Zaokrúhlite na 1 des. miesto

Hodnotenie: Za každú štatistiku môžete získať alikvótnu časť z 10 bodov. Plný počet bodov za relatívnu chybu do 5 %, lineárne klesajúce k 0 bodom.

Podúloha 2 -- Identifikácia (15 bodov)
Pre každý text v test_s2.jsonl určte, či je **ľudský (0)** alebo **strojový (1)**.

Hodnotenie: Váhované F1 skóre s lineárnou normalizáciou. Za F1 pod spodným prahom dostávate minimálne body, za F1 nad horným prahom plný počet.

Podúloha 3 -- Atribúcia (45 bodov)
Pre každý text v test_s3.jsonl určte, ktorý model ho vygeneroval. Odpoveď je jeden z: ModelA, ModelB, ModelC, ModelD.

Trénovacie dáta sú v train_s3.jsonl, validačné v valid_s3.jsonl.

Hodnotenie: Váhované F1 skóre s lineárnou normalizáciou.

Podúloha 4 -- Kontrarozviedka (30 bodov)
Niektoré strojové texty boli **prepísané tak, aby vyzerali ľudsky** -- parafrázované a preštylizované. Tieto texty sa nachádzajú v súbore test_s4.jsonl, zmiešané s pravými ľudskými textami.

Pre každý text v test_s4.jsonl určte, či je **ľudský (0)** alebo **strojový/parafrázovaný (1)**.

Na túto podúlohu **neexistuje trénovacia sada**. Namiesto toho dostanete validačnú sadu valid_s4.jsonl, v ktorej sú texty v pároch -- vždy jeden ľudský a jeden parafrázovaný na tú istú tému. Pozorne si tieto páry preštudujte a nájdite, čo ich odlišuje.

Hodnotenie: Váhované F1 skóre s lineárnou normalizáciou.

Výstupný formát
Odovzdávate jediný súbor vo formáte .csv s troma stĺpcami:

```
1, 0, 47.25
1, 1, 3000
1, 2, 187.4
...
2, 0, 1
2, 1, 0
2, 2, 1
...
3, 0, ModelA
3, 1, ModelD
```

3, 2, ModelB
...
4, 0, 1
4, 1, 0
4, 2, 0
...
Prvý stĺpec je číslo podúlohy (1--4). Druhý stĺpec je index (poradové číslo riadku odpovede na danú podúlohu, počnúc nulou). Tretí stĺpec je odpoveď.

Súbor nemusí obsahovať odpovede na všetky podúlohy. Vzorový súbor s ukážkovými (nesprávnymi) odpoveďami sa nachádza v sample_output.csv.

Dáta

Podúlohy 1 a 2
- train.jsonl -- trénovacia sada s označením label (0 = ľudský, 1 = strojový)
- valid.jsonl -- validačná sada (rovnaký formát ako trénovacia)
- test_s2.jsonl -- testovacia sada (bez anotácií)

Podúloha 3
- train_s3.jsonl -- trénovacia sada s označením model (ModelA, ModelB, ModelC, ModelD)
- valid_s3.jsonl -- validačná sada
- test_s3.jsonl -- testovacia sada (bez anotácií)

Podúloha 4
- valid_s4.jsonl -- validačná sada s pármi textov (ľudský + parafrázovaný na tú istú tému), s označením label a pair_id
- test_s4.jsonl -- testovacia sada (bez anotácií)

Formát dát
Každý riadok je JSON objekt. Príklady:
{"id": "human_0001", "text": "...", "label": 0}
{"id": "v3_StyleA_0001", "text": "...", "label": 1}
{"id": "s3_ModelA_0001", "text": "...", "label": 1, "model": "ModelA"}

Štartovací balíček
V štartovacom balíčku nájdete:
- starter.py -- načítanie dát, výpočet niektorých štatistík, generovanie výstupného CSV
- sample_output.csv -- vzorový výstupný súbor (nesprávne odpovede, správny formát)

Tipy
- Začnite podúlohou 1 -- pri výpočte štatistík si všimnete vzory, ktoré vám pomôžu pri klasifikácii.
- Pre podúlohu 2 skúste najprv jednoduchý TF-IDF + logistická regresia baseline, potom vylepšujte.
- Pre podúlohu 3 sa zamyslite: čím sa líšia rôzne modely? Dĺžka viet? Slovná zásoba? Interpunkcia?
- Pre podúlohu 4 -- ak váš detektor z podúlohy 2 funguje dobre, skúste ho aplikovať aj tu. Ale pozor -- parafrázované texty sú záludnejšie.
- tiktoken knižnica sa dá nainštalovať cez pip install tiktoken. Tokenizér o200k_base sa načíta cez tiktoken.get_encoding("o200k_base").

Veľa zdaru!