Magyar MI Diákolimpia

Korlátozott Generálás

Feladatleírás · Nyári Országos Válogató

2026. május

1. Helyzetjelentés

1969-ben Georges Perec megírta a *La Disparition* című regényét: több mint 300 oldal, egyetlen „e” betű nélkül. Az ilyen szándékosan korlátozott írásművek (lipogramok) máig izgatják a nyelvészeket és az íróműhelyeket.

Egy kísérleti műhely automatizálni szeretné ezt. Egy kis méretű nyelvi modell folytatásokat ír előre megadott promptokhoz, miközben soha nem ejt ki egyetlen tiltott szót. A feladatod, hogy ezt a korlátozott dekódolást implementáld. A modell minden lépésben javasol egy következő tokent (a `logits` formájában), te döntöd el, melyik kerüljön a kimenetbe.

20 kísérlet áll előtted, mindegyik más-más tematikájú prompttal és különböző nehézségű tiltott-szó listával: egyszerű stopszavaktól (`the`, `a`, `an`) a tematikus szóosztály-tiltásokig (egy detektív-szövegben tiltott a teljes erőszakos szókincs, egy programozási leírásban a triviális szakszavak). A kimenetnek hosszúnak, diverznek és természetesen hangzónak kell maradnia, miközben a korlátozást betartja.

A feladatban szereplő szervezetek és események fiktívek (Perec és műve nem).

2. Korlátozott Generálás

A kauzális nyelvi modellek autoregresszív módon állítanak elő szöveget: minden lépésben valószínűségi eloszlást határoznak meg a szókészleten (a *logitek*), majd a következő tokent ezen

eloszlás alapján választják ki. A logitek módosításával, alternatív mintavételezési szabály vagy keresési eljárás alkalmazásával a modell kimenete újratanítás nélkül is korlátozható.

Adott egy kis méretű, előtanított nyelvi modell és egy sor teszteset. Minden teszteset egy kezdő promptból, egy tiltott szavakat tartalmazó listából és a generálható tokenek maximális számából áll. A feladatod, hogy minden egyes generálási lépésre meghatározd a következő token azonosítóját úgy, hogy a végső kimenet egyetlen tiltott szót se tartalmazzon, miközben a hossza, diverzitása és természetessége is megmarad.

Egy szó akkor tekintendő tiltottnak a kimenetben, ha mint különálló szó (szóhatárral elválasztva) előfordul benne, a kis- és nagybetűk megkülönböztetése nélkül. Például ha az `apple` tiltott, akkor az `apple`, `Apple`, `APPLE`, `apple,` és `apple.` előfordulások egyaránt szabályt sértenek. A `pineapple` viszont megengedett, mivel nem önálló `apple` előfordulás.

3. Amit megkapsz

Egy előtanított `Qwen/Qwen3-0.6B-Base` kauzális nyelvi modellt a Hugging Face Transformers könyvtáron keresztül, valamint a hozzá tartozó tokenizert. A modell a kiértékelés során ugyanezekkel a súlyokkal és csomagverziókkal kerül betöltésre, így a kimenet futtatások között determinisztikus.

Egy implementálandó függvénystubot, amelynek szignatúrája:

```
select_next_token(logits, prompt, forbidden, generated_ids) -> int
```

A paraméterek jelentése:

- `logits` – a modell utolsó pozícióra adott kimeneti logitjai (méret: `[vocab_size]`),
- `prompt` – a kiindulási szöveg,
- `forbidden` – a tiltott szavak listája,
- `generated_ids` – az eddig kiválasztott tokenek azonosítóinak listája.

A visszatérési érték a következő token azonosítója egészként, 0 és `vocab_size`−1 között.

A mellékelt notebookban rendelkezésre áll a teljes értékelő keretrendszer (`generate_constrained`, `score_case`, `run_tests`), egy `would_violate` segédfüggvény, amely az értékeléssel azonos szabályok szerint jelzi, hogy egy adott részeredmény tartalmaz-e tiltott szót, valamint egy `make_submission()` segédfüggvény, amely a megoldott `select_next_token` alapján előállítja a beadandó `submission.csv`-t.

A `test_cases.json` fájl tartalmazza a hivatalos tesztkészletet (jelenleg **20 teszteset**) egy lista formájában, amelynek minden eleme egy teszteset az `id`, `prompt`, `forbidden` és `max_new_tokens` mezőkkel. A pontozás kizárólag ezen a tesztkészleten történik. Ugyanezt a fájlt kell a generáláshoz is használni.

4. Amit beadsz

Egyetlen CSV fájlt `id,output,fluency,tokens` fejléccel, amelyben pontosan annyi adatsor van, ahány teszteset a `test_cases.json` fájlban szerepel. Az oszlopok jelentése:

- `id` – a `test_cases.json` megfelelő bejegyzésére hivatkozik.
- `output` – a generált szöveg (a prompt nélkül).
- `fluency` – a $\Phi = \exp(\overline{\log p_\theta}/5)$ érték a `Qwen3-0.6B-Base` modellből, a teljes `(prompt + output)` szövegen kiszámolva. A notebook `score_case` függvénye automatikusan számolja és írja be ezt az értéket.
- `tokens` – a `tokenizer.encode(output)` hossza.

A fájl szabványos CSV-formátumot követ (RFC 4180): a vesszőt, idézőjelet vagy sortörést tartalmazó mezőket idézőjelbe kell tenni, az idézőjeleket pedig duplázni szükséges. A beadáshoz a mellékelt `make_submission()` segédfüggvény használata ajánlott. Ez a fenti szabályokat automatikusan kezeli, és a kimenetet UTF-8 kódolással írja ki.

A `fluency` és `tokens` mezőket a notebook `make_submission()` függvénye számolja és írja be. Ezeket **nem szabad kézzel módosítani**: a kiértékelő a beadott értékeket felülbírálja.

A kimenetet kizárólag a feladatban megadott `Qwen/Qwen3-0.6B-Base` modellel szabad előállítani. Bármilyen más generatív modell használata azonnali kizárást von maga után. A modell saját dekódolási stratégiája tetszőlegesen megengedett, amennyiben kizárólag a megadott modell logitjeit használja.

5. A feladat pontozása

Jelölje $\mathcal{T}$ a tesztkészletet, és egy adott teszteset esetén p a promptot, $\mathcal{F}$ a tiltott szavak halmazát, M a megengedett tokenmaximumot, y pedig a generált kimeneti szöveget. A teszteset pontszáma az alábbiak szerint adódik:

- ha y tartalmaz tiltott szót: **0 pont**,
- ha y üres: **0 pont**,
- egyébként: $\text{pont}(y) = L \cdot \sqrt{\text{distinct}_2} \cdot \Phi$, ahol L a hosszúsági arány, distinct_2 a bigram-diverzitás, Φ pedig a fluencia.

Per-teszteset maximum: 1,0 (ha $L = \sqrt{\text{distinct}_2} = \Phi = 1$).

Pontozási tényezők. A három tényezőt az alábbi képletek határozzák meg.

Tényező	Képlet
L	$\min\left(1{,}0,\ \left(\frac{\text{tokenek száma}}{M}\right)^{1{,}5}\right)$
distinct_2	$\frac{\lvert\{\text{egyedi szóbigramok}\}\rvert}{\lvert\{\text{összes szóbigram}\}\rvert}$
Φ	$\exp\left(\frac{\overline{\log p_\theta}}{5}\right)$

A hosszúsági arány 1,5-ös hatványa erősebben bünteti a rövid kimeneteket, az $L \leq 1{,}0$ felső korlát pedig megakadályozza a tokenmaximumon túli díjazást. A distinct$_2$-t a kimenet kisbetűsített, `\w+` mintával szavakra bontott alakja fölötti egyedi bigramok arányaként számoljuk, így az ismétlődő szófűzések pontlevonással járnak. A fluenciát a modell saját átlagos token-log-valószínűsége adja, ezért a természetellenes, valószínűtlen folytatások szintén csökkentik a pontszámot.

Végső pontszám. A tesztesetenkénti pontszámok *átlaga* $\times 100$ adja a beadás végleges eredményét. A skála így **[0, 100]** bármely tesztkészlet-mérettel.

Nyilvános és privát részhalmaz. A 20 teszteset két részre van osztva: **6 nyilvános eset** (a verseny alatt látható score-t ezekre számoljuk) és **14 privát eset** (csak a verseny végén derül ki, ez dönti el a végleges helyezést). A `submission.csv`-be mind a 20 teszteset kimenete kell, a felosztás csak a pontszám megjelenítésében jelenik meg. A felosztás célja, hogy a verseny közbeni ranglistára való túl-tuningolás (*leaderboard-grinding*) gyengüljön.

6. Technikai tudnivalók

A feladat megoldásához a mellékelt `.ipynb` notebook és a `test_cases.json` fájl áll rendelkezésre. A notebook tartalmazza a modell és a tokenizer betöltését, az értékelő keretrendszert, valamint a `would_violate` és `make_submission` segédfüggvényeket. A megoldó feladata a `select_next_token` függvény implementálása, a notebook lefuttatása a `test_cases.json` tesztkészleten, és az így előálló `submission.csv` beadása.

Csak a kijelölt cellát (`select_next_token`) szabad módosítani. A modellt, a generálási loopot és a pontozási keretrendszer egyetlen függvényét sem szabad átírni. A beadott notebook poszt-hoc auditra kerül, és bármilyen pontozást megkerülő manipuláció kizárást von maga után.

T4-es GPU ajánlott. CPU-n is fut, de jelentősen lassabban.

7. Hasznos források

- PyTorch dokumentáció
- Hugging Face Transformers dokumentáció
- Qwen3-0.6B-Base modellkártya