Մեքենայի ուրվականը

- **Ժամանակի սահմանափակում.** 10 րոպե
- **Baseline միավոր.** 28.6
- **Միջավայր.** մեկ GPU (≈16 ԳԲ VRAM), առանց ինտերնետի
- **Լուծման չափը.** `solution.ipynb` ≤ 20 ՄԲ
- **Հիշողություն.** 5 ԳԲ
- **Նախավարժեցված մոդելներ.** միայն **bge-base-en-v1.5** — տեքստային **encoder** (embedding model)։

Առաջադրանք

Ղազախստանի ազգային արխիվում տարօրինակ բաներ են տեղի ունենում։ Գրադարանավարներն ասում են, որ որոշ գրքեր նախկինում այլ կերպ էին ավարտվում, բայց ոչ ոք չի կարող դա ապացուցել. յուրաքանչյուր օրինակ նույնն է, և յուրաքանչյուր պատմություն դեռևս իմաստալից է։ Դուք, որպես ԱԻ (AI) հետազոտող, հրավիրված եք գտնելու այդ փոփոխությունները։

Տեքստի հատվածը սկսվում է որպես մարդու կողմից գրված տեքստ և, որոշակի պահի, աննկատորեն փոխարկվում է լեզվական մոդելի կողմից գեներացված շարունակության։ Ամբողջությամբ կարդալիս այն թվում է մեկ կապակցված տեքստ, բայց ինչ-որ տեղ մեջտեղում հեղինակը մարդուց փոխվում է մեքենայի։ Ձեր առաջադրանքն է **գտնել այդ փոխարկումը. նիշի ինդեքսը, որտեղ ավարտվում է մարդկային մասը և սկսվում է մեքենայական մասը**։

Յուրաքանչյուր նմուշ մեկ տող է՝ `text`։ Այնտեղ կա ճիշտ մեկ սահման։ Դրանից առաջ ամեն ինչ մարդու կողմից է, իսկ դրանից սկսած ամեն ինչ մեքենայի կողմից է գեներացված։

Dataset

Սովորական տեքստային (plain-text) անգլերեն հատվածներ, յուրաքանչյուրը՝ մեկ սահմանով։

- **Մաս A** (սահմանից առաջ). մարդու կողմից գրված տեքստ։
- **Մաս B** (սահմանից սկսած). լեզվական մոդելի կողմից ստեղծված շարունակություն՝ հիմնված Մաս A-ի վրա։
- Յուրաքանչյուր կողմն առնվազն 180 բառ է. ընդհանուր երկարությունը ~500–800 բառ է։
- **`boundary_char_index`**-ը նիշի շեղումն (character offset) է, որտեղ ավարտվում է Մաս A-ն. `text[:boundary_char_index]`-ը մարդկային մասն է, իսկ `text[boundary_char_index:].lstrip()`-ը՝ մեքենայական մասը։

Ինչ եք ստանում դուք

Դուք ստանում եք **երկու թղթապանակ**.

Թղթապանակ	Նմուշներ	`answers.jsonl`?	Օգտագործեք այն
`dataset/train/`	1,221	ներառված է	վարժեցնելու / fine-tune անելու ձեր մեթոդը
`dataset/test_public/`	380	ներառված է (dev պատճեն)	աշխատեցնելու ձեր pipeline-ը և տեղում ինքնագնահատվելու համար

Գնահատման պահին ձեր `dataset/test_public/` թղթապանակը **փոխարինվում է թաքնված գնահատման տվյալների բազմությամբ (evaluation set)**։ Այն ունի նույն ձևաչափը, բայց **առանց `answers.jsonl`-ի**։ Ձեր notebook-ը վերագործարկվում է դրա վրա, և դրա ստեղծած `answers.jsonl`-ը գնահատվում է։

- Հանրային leaderboard-ն օգտագործում է թաքնված **test_leaderboard_a** բազմությունը (380 նմուշ)։
- Վերջնական վարկանիշային աղյուսակն օգտագործում է թաքնված **test_leaderboard_b** բազմությունը (380 նմուշ)։

Բոլոր երեք գնահատման բազմությունները նույն չափի են և վերցված են նույն բաշխումից, ինչ `train`-ը, ուստի ձեր տեղական `dataset/test_public/` միավորը ողջամիտ գնահատական է leaderboard-ի ձեր միավորի համար։

Սկավառակի վրա ձևաչափը

```
dataset/train/data.jsonl      # one JSON object per line: {"id": "...", "text": "..."}
dataset/train/answers.jsonl   # {"id": "...", "boundary_char_index": 1842}
dataset/test_public/data.jsonl       # {"id": "...", "text": "..."}
dataset/test_public/answers.jsonl    # dev copy only — ABSENT in the hidden grading set
```

- id-ները `answers.jsonl`-ում համապատասխանում են id-ներին `data.jsonl`-ում։
- `dataset/train/`-ը (պատասխաններով) հասանելի է ամեն անգամ, երբ դուք վարժեցնում կամ fine-tune եք անում։

Ելքային ձևաչափ (submission format)

Դուք ուղարկում եք **մեկ notebook, որը պետք է անվանված լինի `solution.ipynb`**: Ֆայլի հենց այս անվանումն է պահանջվում: Ցանկացած այլ բան մերժվում է՝ առանց աշխատեցվելու:

Ձեր notebook-ը պետք է **կարդա `dataset/test_public/data.jsonl`**-ը և գրի մեկ ֆայլ՝ **`answers.jsonl`**, ռեպոզիտորիայի արմատային թղթապանակում — յուրաքանչյուր տողում մեկ JSON օբյեկտ, որը արտապատկերում է յուրաքանչյուր sample id-ն ձեր կանխատեսած սահմանային նիշի ինդեքսին.

```
{"id": "example_000123", "boundary_char_index": 1790}
{"id": "example_000124", "boundary_char_index": 2450}
```

- `boundary_char_index`-ը պետք է լինի **ամբողջ թիվ `[0, len(text)]` միջակայքում**:
- `dataset/test_public/data.jsonl`-ի յուրաքանչյուր id պետք է հանդիպի ճիշտ մեկ անգամ: `answers.jsonl`-ից բացակայող նմուշը (կամ ոչ ամբողջ թիվ / միջակայքից դուրս արժեք ունեցող) այդ նմուշի համար ստանում է 0 միավոր:

Գնահատում

Յուրաքանչյուր նմուշի համար ենթադրենք `p`-ը ձեր կանխատեսած ինդեքսն է, իսկ `t`-ը՝ իսկական սահմանը: Յուրաքանչյուր նմուշի միավորը նվազում է էքսպոնենցիալ կերպով՝ կախված նիշերի հեռավորությունից.

$$\text{score} = \exp\left(-\frac{|p-t|}{\tau}\right) \in (0,1], \text{ where } \tau = 100.$$

Սա հանգեցնում է միավորի հետևյալ վարքագծին.

- **=1.0** — սահմանի ճշգրիտ նիշ;
- **≈0.78** — 25 նիշի շեղում; - **≈0.61** — 50 նիշի շեղում;
- **≈0.37** — 100 նիշի շեղում;
- **≈0.01** — 500 նիշի շեղում:

Վերջնական միավորը տվյալ բազմության բոլոր նմուշների ըստ-նմուշի միավորների **միջինն է** (ներկայացված 0–100 սանդղակով): Մետրիկան պարգևատրում է *մոտ* լինելու համար, ոչ միայն ճշգրիտ:

Սահմանափակումներ

- **Միջավայր.** մեկ GPU (≈16 ԳԲ VRAM), գնահատման պահին առանց ինտերնետի — թույլատրված մոդելը (ստորև) արդեն տրամադրված է: **Իրական ժամանակի (Wall-clock) բյուջե. 10 րոպե** ամբողջ աշխատանքի համար — սա պետք է ծածկի գնահատման պահին ձեր կատարած ցանկացած վարժեցում / fine-tuning **գումարած** ինֆերենսը (inference) գնահատման բազմության վրա:
- **Թույլատրված նախավարժեցված մոդել** — սա սպառիչ է, որևէ այլ նախավարժեցված կշիռներ (pretrained weights) չեն կարող օգտագործվել: Այն **նախապես տրամադրված է միջավայրում** (բեռնեք այն սովորական ձևով, օրինակ՝ `from_pretrained`, գնահատման պահին ինտերնետ չկա).
    - **bge-base-en-v1.5** — 110M-պարամետրով տեքստային **encoder** (embedding model): Այն ստեղծում է նախադասության/հատվածի embedding-ներ, այն գեներատիվ լեզվական մոդել չէ: Դուք կարող եք օգտագործել այն **ինչպես որ կա (սառեցված հատկանիշներ) կամ fine-**

tune անել այն `train` բազմության վրա (ամբողջական fine-tuning-ը տեղավորվում է 16 ԳԲ / 10-րոպեանոց բյուջեի մեջ)։

- Դասական / վիճակագրական գործիքներն անսահմանափակ են. դուք կարող եք կառուցել ցանկացած հատկանիշների վրա հիմնված (feature-based) մոդել (օրինակ՝ scikit-learn դասակարգիչներ կամ ռեգրեսորներ) ձեր իսկ հաշվարկած embedding հատկանիշների վրա։ *Նախավարժեցված խորը ուսուցման (deep-learning) կշիռները* սահմանափակված են միայն վերը նշված ցանկով։