Գտնել կարգը

- **Ժամանակի սահմանափակում.** 10 րոպե
- **Միջավայր.** մեկ GPU (≈16 ԳԲ VRAM), առանց ինտերնետի
- **Լուծման չափս.** `solution.ipynb` ≤ 1 ՄԲ
- **Հիշողություն.** 5 ԳԲ

Խնդիրը

Ձեզ տրված են անգլերեն խոսակցական երկխոսություններ երկու մասնակիցների՝ *խոսնակ A*-ի (*Speaker A*) և *խոսնակ B*-ի (*Speaker B*) միջև: Յուրաքանչյուր երկխոսություն բաժանված է խոսնակների ռեպլիկների (speaker turns), ընդ որում յուրաքանչյուր ռեպլիկ պարունակում է միայն մեկ խոսնակի խոսք: Ամեն ռեպլիկ պահված է որպես առանձին `.wav` աուդիո ֆայլ, ուստի ամբողջական երկխոսությունը ներկայացված է `.wav` ֆայլերի բազմությամբ՝ մեկ ֆայլ յուրաքանչյուր ռեպլիկի համար:

Ցավոք, ռեպլիկները պատահականորեն խառնվել են, ուստի զրույցն այլևս իմաստ չի արտահայտում: `chunk_{k}.wav` ֆայլի անվանման մեջ `k`-ը վերաբերում է խառնված բազմության k-րդ կտորին (chunk), այլ ոչ թե բնօրինակ երկխոսության k-րդ ռեպլիկին:

!! Ձեր խնդիրն է վերականգնել զրույցի սկզբնական ժամանակագրական կարգը: (Your task is to reconstruct the original chronological order of the conversation.)

Dataset

Յուրաքանչյուր երկխոսություն պարունակում է `n` աուդիո ֆայլեր՝ `chunk_0.wav`, `chunk_1.wav`, …, `chunk_{n-1}.wav` անվանումներով։ Կտորները (chunks) առանձին ռեպլիկներն են։ Ֆայլերի անվանումները համապատասխանում են միայն խառնված կարգին։ Դրանք չեն նշում, թե կտորը որտեղ է գտնվում սկզբնական զրույցում։ Յուրաքանչյուր երկխոսություն ունի 7–20 կտոր, մոնո, 44.1 կՀց (կարող եք resample անել)։

`prefix.json`-ը պարունակում է յուրաքանչյուր երկխոսության առաջին երկու կտորների ֆայլերի անվանումների ինդեքսները։ Սա որոշում է երկխոսության ճշգրիտ սկիզբը և վերացնում է անորոշությունը զրույցը ուղիղ կամ հակառակ հերթականությամբ կարդալու միջև։

Օրինակ՝ `11: [7, 12]` նշանակում է, որ 11-րդ երկխոսության առաջին և երկրորդ ռեպլիկները համապատասխանաբար `chunk_7.wav` և `chunk_12.wav` ֆայլերն են։

Ինչ եք ստանում

Դուք ստանում եք **երկու թղթապանակ (folder)՝ նույն ձևաչափով**.

Folder	Երկխոսություններ	`answers.json`?	Ինչպես Օգտագործեք այն
`dataset/train/`	1,288	✅ ներառված է	մոդելը train / fine-tune անելու համար
`dataset/test_public/`	100	✅ ներառված է	pipeline-ը աշխատեցնելու և տեղում (locally) ինքնագնահատվելու համար

Գնահատման ընթացքում ձեր `dataset/test_public/` թղթապանակը թափանցիկորեն փոխարինվում է `hidden evaluation set`-ով (`test_leaderboard_a`՝ հանրային leaderboard-ի համար, և `test_leaderboard_b`՝ վերջնական leaderboard-ի համար) — դրանք ունեն նույն չափսը և ձևաչափը, ինչ `dataset/test_public/`-ը, բայց առանց `answers.json`-ի:

Ձեր notebook-ը նորից աշխատեցվում է այդ տվյալների վրա, և դրա գեներացրած `answers.json` ֆայլն օգտագործվում է գնահատման համար: Համակարգում պահվող test երկխոսությունները նույն բաշխումից են, ինչ `train`-ը, ուստի ձեր local `test_public` միավորը հավաստի նախադիտում է:

Թղթապանակների կառուցվածքը

```
dataset/train/
    prefix.json  # {dialogue_id: [first_idx, second_idx]} filename index
    answers.json  # {dialogue_id: P}  ground-truth order (rank convention)
    /
        chunk_0.wav
        ...
        chunk_{n-1}.wav

dataset/test_public/
    prefix.json
    answers.json     # present only in the development copy
    /
        chunk_0.wav
        ...
        chunk_{n-1}.wav
```

Արդյունք (Output)

Յուրաքանչյուր երկխոսության համար որոշեք դրա աուդիո կտորների սկզբնական ժամանակագրական կարգը։ Ձեր կանխատեսումը պետք է լինի `{0, 1, …, n−1}`-ի `P` տեղափոխություն (permutation), որտեղ `P[i]`-ը `chunk_i.wav`-ի կանխատեսված ժամանակագրական դիրքն է (0 = առաջին)։

Ձեր ելքային `answers.json` ֆայլը պետք է արտապատկերի (map) յուրաքանչյուր երկխոսության ID-ն իր կանխատեսված տեղափոխությանը.

```
{
  "17": [2, 0, 1],
  "42": [0, 1],
  "108": [3, 1, 0, 4, 2, 5]
}
```

Օրինակ

Երկխոսությունն ունի 3 խառնված կտոր `chunk_0, chunk_1, chunk_2.`

խառնված կտոր	արտասանված բովանդակություն	իրական դիրք (կարգ)
`chunk_0.wav`	*"Մի մտահոգվիր — ես հետո քեզ կուղարկեմ նշումները։"*	2 (վերջին)
`chunk_1.wav`	*"Հե՛յ, գալի՞ս ես ժամը երեքի հանդիպմանը։"*	0 (առաջին)
`chunk_2.wav`	*"Չեմ կարող — այդ ժամին ատամնաբույժի այցելություն ունեմ։"*	1

Իրական կարգն է **chunk_1 → chunk_2 → chunk_0**, ուստի `P = [2, 0, 1]`, իսկ `prefix.json`-ը պարունակում է `[1, 2]`։

⚠ **P-ն պետք է լինի իսկական տեղափոխություն (permutation).** երկարությունը n, 0-ից ինդեքսավորված, յուրաքանչյուր արժեք ճիշտ մեկ անգամ։ Կրկնությունները, բաց թողնված արժեքները կամ տիրույթից դուրս գրառումները (օրինակ՝ 1-ից ինդեքսավորված) տվյալ երկխոսության համար գնահատվում են 0 միավոր։ Ֆայլից բացակայող երկխոսությունը գնահատվում է 0։ Սխալ ձևավորված կամ ոչ JSON ֆայլը մերժվում է։

Գնահատում

Այս խնդրի համար գնահատումը **զույգերով կարգավորման ճշգրտությունն է (pairwise ordering accuracy)**: Այն ստուգում է կտորների յուրաքանչյուր զույգ և հարցնում. *երկուսից ո՞րը պետք է լինի առաջինը:* Զույգը ճիշտ է, եթե ձեր կանխատեսումը տալիս է նույն պատասխանը, ինչ իրականությունը (ground truth): `n` կտորներով երկխոսության համար կան

$$M = n(n-1)/2$$

զույգեր. թող `I`-ը լինի ինվերսիաների (inversions) քանակը — զույգեր, որոնք կարգավորված են իրականությունից (ground truth) տարբերվող ձևով.

$$\text{score} = 1 - \frac{I}{M} \in [0, 1]$$

i **Վերջնական միավորը split-ի բոլոր երկխոսությունների ըստ երկխոսության միավորների միջինն է:**

Թույլատրված մոդելներ

Այս խնդիրը լուծելու համար դուք կարող եք օգտագործել միայն հետևյալ pre-trained մոդելները՝ ինչպես training-ի, այնպես էլ evaluation-ի ընթացքում: Այս բոլոր մոդելներն արդեն ներբեռնված են և հասանելի են միջավայրում: Դուք կարող եք տեսնել դրանց օգտագործման օրինակները `solution.ipynb` baseline notebook-ում: Խնդրում ենք նկատի ունենալ, որ դուք չեք կարող օգտագործել որևէ այլ մոդել, և ձեր ծրագիրը չունի ինտերնետի հասանելիություն:

- **Խոսքի ներկայացումներ (Speech representations). wav2vec 2.0**: **Whisper encoder**-ը նույնպես կարող է օգտագործվել որպես feature extractor: [wav2vec model card]
- **Խոսքի ավտոմատ ճանաչում (ASR). OpenAI Whisper** (ցանկացած չափսի): [Whisper model card]
- **Լեզվական մոդել (Language model). Qwen2.5-0.5B**, որը կարող է օգտագործվել կամ zero-shot, կամ fine-tuneտրամադրվածtrain՝ split-ի վրա: [Qwen model card] Նկատի ունեցեք, որ 10 րոպեի սահմանափակումը պետք է ներառի ցանկացած training կամ fine-tuning, որը դուք անում եք գնահատման (grade time) պահին, պլյուս inference-ը evaluation բազմության վրա:

Ինչպես ուղարկել (How to submit)

- Բացեք `solution.ipynb`-ը և աշխատեցրեք բոլոր բջիջները (cells): Համոզվեք, որ այն աշխատանքային folderում գրում է `answers.json` ֆայլը՝

`dataset/test_public/`-ի յուրաքանչյուր երկխոսության համար (100 երկխոսություն) with the correct permutations: Գնահատման ընթացքում notebook-ը վերընթացվում է թաքնված test բազմության վրա, և դրա արտադրած `answers.json`-ը գնահատվում է:

- Բարելավեք լուծումը, եթե ցանկանում եք. baseline-ը զուտ աշխատեցնում է pipeline-ը:
- Բացեք Git ներդիրը JupyterLab-ի ձախ կողմնային վահանակում (sidebar):
- **Stage** արեք `solution.ipynb`-ը (դրա կողքի + պատկերակը):
- Մուտքագրեք commit-ի հաղորդագրությունը և սեղմեք **Commit**:
- Սեղմեք վերև սլաքով ամպի պատկերակը՝ push անելու համար:
- Վերադարձեք այս Մրցույթի էջը և սեղմեք **Submit**:

Ուղարկեք ճիշտ մեկ ֆայլ՝ `solution.ipynb` անվանումով: