DATA SCIENCE AZ IT BIZTONSÁGBAN

4. forduló

SOPHOS

A kategória támogatója: SOPHOS

Ismertető a feladathoz

A 4. forduló után elérhetőek lesznek a helyezések %-os formában: azaz kiderül, hogy a kategóriában a versenyzők TOP 20% - 40% -60% -ához tartozol-e!

Szeretnénk rá felhívni figyelmedet, hogy a játék nem Forma-1-es verseny! Ha a gyorsaságod miatt kilököd a rendesen haladó versenyzőket, kizárást vonhat maga után!

4.forduló

Ebben a fordulóban state of the art (SOTA) algoritmusok elméletét nézzük át a Natural Language Processing (NLP) terén.

Hatalmas input esetén minden egyes szó elkódolása nem egy járható út, a hatalmas memória- és számítási igény miatt.

Mig a karakterszintű tokenizáció nagyon egyszerű és kis memória lenyomatú, jelentős performancia veszteséggel járhat, mivel a betűk nem tartják meg a kontextusukat.

A transformer modellekhez egy köztes tokenizációs megoldást választottak, az úgynevezett "subword" tokenizációt.

A subword tokenizáció különböző variációnak a lényege, hogy a gyakran előforduló szavakat elkódolja, a ritkábbakat viszont szétdarabolja különböző "subword"-ökre. A forduló első felében ezeket a tokenizációs stratégiákat nézzük át közelebbről. A forduló fennmaradó részében a transformer architektúráról nézünk meg pár kérdést, amit az ötödik fordulóban gyakorlati szemszögből is érintünk.

Felhasznált idő: 00:00/40:00 **Elért pontszám:** 0/16

1. feladat 0/2 pont

Melyik igaz a következő állítások közül?

Válasz

- (●) A Byte Pair Encoding (BPE) a "merge" lépésnél megszámolja az összes lehetséges szimbólumpár gyakoriságát és azt a szimbólumpárt választja, ami a legtöbbször fordul elő
  Ez a válasz helyes, de nem jelölted meg.
- ( ) A WordPiece tokenizer azokat a szimbólumpárokat fűzi össze a "merge" lépésnél, amelyeknek a valószínűsége osztva az első szimbólum valószínűségével és a második szimbólum valószínűségével a legnagyobb az összes szimbólumpáré közül
- ( ) A Sentence tokenizer feltételezi, hogy a bemeneti szöveg szószeparátorként szóközöket használ

Magyarázat

https://huggingface.co/docs/transformers/tokenizer_summary

2. feladat 0/1 pont

Mi volt a következő distilbert-base-uncased kódolás bemenő adata?

[101, 12669, 13759, 4674, 4570, 1012, 15876, 1013, 5736, 20255, 2401, 5714, 102]

Válaszok

A helyes válasz:

megmerettetes.hu/kategoriaim

[CLS] megmerettetes. hu / kategoriaim [SEP]

Magyarázat

Kedves Versenyzők!

A [ ]-t nem ismeri fel a kiértékelő algoritmus, a javítás folyamatban van!

Köszönjük a türelmet!

```
from transformers import AutoTokenizer
MODEL = 'distilbert-base-uncased'
tokenizer = AutoTokenizer.from_pretrained(MODEL)
tokenizer.convert_ids_to_tokens([101, 12669, 13759, 4674, 4570, 1012, 15876, 1013, 5736, 20255, 2401, 5714, 1
```

3. feladat 0/3 pont

A pretokenizálás után a következő tokeneket és előfordulásokat kaptuk bemenetként:

("we", 10), ("fun", 12), ("and", 7), ("run", 4), ("love", 5)

BPE esetén a harmadik merge lépésnél melyik subword-del bővül a szótárunk?

Válasz

A helyes válasz:

we

Magyarázat

Input: ("we", 10), ("fun", 12), ("and", 7), ("run", 4), ("love", 6)

Input vocab: ("w" "e", 10), ("f" "u" "n", 12), ("a" "n" "d", 7), ("r" "u" "n", 4), ("w" "a" "n" "t", 6)

1. lépés; ("w" "e", 10), ("f" "un", 12), ("a" "n" "d", 7), ("r" "un", 4), ("w" "a" "n" "t", 6)
2. lépés; ("w" "e", 10), ("f" "un", 12), ("an" "d", 7), ("r" "un", 4), ("w" "an" "t", 6)
3. lépés; ("w" "e", 10), ("fun", 12), ("an" "d", 7), ("r" "un", 4), ("w" "an" "t", 6)

4. feladat 0/1 pont

Melyik intézmény munkatársai publikálták a Transformer architektúrát?

Válasz

- Hugging Face
- OpenAI
- Google
  Ez a válasz helyes, de nem jelölted meg.
- MIT

Magyarázat

https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)

https://arxiv.org/abs/1706.03762

5. feladat 0/3 pont

Melyek a megfelelő opciók a Natural Language Processing (NLP) finomhangolásához?

Válasz

- Adapter modulokat adunk egy pre-trainelt modellhez, és csak az adapter modulokat finomhangoljuk
- Minden paraméterét finomhangoljuk a pre-trainelt modellnek

- ( ) Csak a "bias" paramétereit finomhangoljuk a modellnek.
- (•) Mind
  Ez a válasz helyes, de nem jelölted meg.

Magyarázat

Az alábbi cikkben megtalálhatjuk a hivatkozást mindhárom módszerre, illetve láthatjuk az egymáshoz hasonlitott hatékonyságukat is.

https://arxiv.org/pdf/2110.04366.pdf

6. feladat 0/3 pont

Az alábbi állítások közül melyek **igazak** az attention mechanizmusra?

Válasz

- (•) Lehetővé teszi, hogy a modell különböző hangsúlyt fektessen az adat különböző részeire
  Ez a válasz helyes, de nem jelölted meg.
- ( ) Tényleges gyakorlatban tetszőleges hosszú stringen végre lehet hajtani
- ( ) L hosszú bementen L^3 időt és memóriát vesz igénybe
- ( ) Megoldja az őt megelőző seq2seq modellek vanishing/exploding gradiens problémáját

Magyarázat

https://jalammar.github.io/illustrated-transformer/

Kedves Versenyzők!

A "Megoldja az őt megelőző seq2seq modellek vanishing/exploding gradiens problémáját" válaszlehetőséget kivettük a helyes válaszok köréből, van, ahol fenn állhat a vanishing gradient probléma (RNN+tanh+Attention), vagy bármilyen, elég mély (pl Sequential Dense) architektúra, tanh/sigmoid aktivációs függvényekkel, ami valahol fel van szerelve attention mechanizmussal.

7. feladat 0/3 pont

Az alábbi állítások közül melyek **igazak** a multi-head attention-re?

Válaszok

- [x] A segítségével a modell még differenciáltabban tud az adat különböző pozícióira fókuszálni
  Ez a válasz helyes, de nem jelölted meg.
- [x] Több reprezentációs alteret biztosít az attention layer számára
  Ez a válasz helyes, de nem jelölted meg.
- [ ] Minden attention head-et ugyanazzal a seed-del inicializálunk
- [x] A feed-forward layer előtt konkatenáljuk az attention head-ek kimenetét
  Ez a válasz helyes, de nem jelölted meg.

Magyarázat

https://jalammar.github.io/illustrated-transformer/