DATA SCIENCE AZ IT BIZTONSÁGBAN

5. forduló

SOPHOS

A kategória támogatója: SOPHOS

Ismertető a feladathoz

Ebben a fordulóban egy Transformer modellt fogunk URL klasszifikációhoz használni, a korábbi fordulókban már megismert adathalmazon. A forduló második felében megismerkedünk a koszinusz hasonlósággal és megvizsgáljuk, hogy a Transformer architektúra mennyivel robosztusabb mint a korábbi n-gram + logistic regression megközelítés.

Időspórolás érdekében finomhangoltunk előre egy transformer modellt.

Az alábbi linken elérhető:

https://oitm-competition.s3.eu-west-2.amazonaws.com/round5/urlbert.tar.gz

Felhasznált idő: 00:00/40:00

Elért pontszám: 0/16

1. feladat 0/1 pont

Mi az általunk finomhangolt base modell arhitektúrájának a neve?

Válaszok

A helyes válasz:

AlbertForSequenceClassification

Albert

albert-base-v1

albert-base-v2

albert

Albert For Sequence Classification

Magyarázat

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification
MODEL = './urlbert'

model = AutoModelForSequenceClassification.from_pretrained(MODEL)
model.config.architectures
```

2. feladat 0/2 pont

A modellünk 45MB diszken. Ha megtartjuk a paraméter szám / diszk méret arányt, mekkora volt 2021 legnagyobb modellje? Terrabyte-ra kerekítve. :-)

Válasz

A helyes válasz:

2

Magyarázat

https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/

```
sum(p.numel() for p in model.parameters())
11685122

( 45 / 11685122 )* 530000000000
```

3. feladat 0/2 pont

Mennyi a maliciousness valószínűsége a "rottentomatoes[.]com/uwbrvz8quk/a[.]exe" oldalnak a pre-trainelt modellünkkel, három tizedesjegyre?

Válaszok

A helyes válasz:

0.999

.999

99.9%

,999

Magyarázat

```
tokenizer = AutoTokenizer.from_pretrained('./urlbert')
model = AutoModelForSequenceClassification.from_pretrained('./urlbert')
test_encoding = tokenizer(['rottentomatoes[.]com/uwbrvz8quk/a[.]exe'], return_tensors='pt', truncation=True,

logits = model(**test_encoding)
torch.nn.functional.softmax(logits.logits, dim=-1)[:, 1].cpu().detach().numpy()
```

4. feladat 0/2 pont

Mi az oka annak, hogy a transformer modellünk jobban teljesít, mint az előző megközelítések?

Válaszok

- [x] A neurális háló kifejezőereje nagyobb, mint a Logistic regression-é
  Ez a válasz helyes, de nem jelölted meg.
- [x] Transfer learning segítségével plusz információ áll a modell rendelkezésére
  Ez a válasz helyes, de nem jelölted meg.
- [x] A "word embedding"-ek kifejezik a szemantikus hasonlóságot szavak között, a megelőző tokenizációkkal ellentétben
  Ez a válasz helyes, de nem jelölted meg.
- [ ] A modell kártékony/ártalmatlan URL adathalmazon volt pre-trainelve

Magyarázat

Az ALBERT, ugyanúgy mint a BERT, az angol Wikipédián és a BookCorpuson lett pre-trainelve.

A többi állítás igaz.

5. feladat 0/2 pont

Az alábbiak közül mi igaz a koszinusz hasonlóságra?

Válaszok

- [x] A koszinus hasonlóság két vektor által közbezárt szög koszinusza
  Ez a válasz helyes, de nem jelölted meg.
- [ ] Jelentősen eltérő szavak esetén 1 közeli értéket ad a koszinus hasonlóság

- [x] Azért előnyös a koszinus hasonlóság használata, mert a vektorok mérete nem befolyásolja
  Ez a válasz helyes, de nem jelölted meg.
- [x] A koszinusz hasonlóság a [-1, 1] intervallumon vehet fel értékeket
  Ez a válasz helyes, de nem jelölted meg.

Magyarázat

https://en.wikipedia.org/wiki/Cosine_similarity

6. feladat 0/2 pont

Mi a koszinus hasonlóság az alábbi két URL korábban használt n-gram kódolása között, két tizedesjegyig?

A korábban tanított ngram tokenizer megtalálható az alábbi linken:

https://oitm-competition.s3.eu-west-2.amazonaws.com/round5/ngram_vectorizer.pkl

URL1 = 'google[.]com/NJttRcKGdAjwfU/'

URL2 = 'google[.]com/yYrzmJgjpKHHe/'

Válasz

A helyes válasz:

0.77

Magyarázat

```
import pickle
from scipy import spatial

ngram_vectorizer = pickle.load(open('ngram_vectorizer.pkl', 'rb'))

embeddings1 = ngram_vectorizer.transform([URL1]).toarray()
embeddings2 = ngram_vectorizer.transform([URL2]).toarray()

result = 1 - spatial.distance.cosine(embeddings1, embeddings2)
```

7. feladat 0/5 pont

Tekintsünk a finomhangolt transformer modell utolsó hidden state-jének mean pool-olt értékét a bemeneti string egy látens térben elhelyezett reprezentációjának. (embedding)

Mi a koszinus hasonlóság az alábbi két URL transformer által generált repreztációja között, két tizedesjegyig?

URL1 = 'google[.]com/NJttRcKGdAjwfU/'

URL2 = 'google[.]com/yYrzmJgjpKHHe/'

Válaszok

A helyes válasz:

0.95

.95

Magyarázat

https://towardsdatascience.com/bert-for-measuring-text-similarity-eec91c6bf9e1

```
from sentence_transformers import SentenceTransformer
model = SentenceTransformer('./urlbert')
embeddings1 = model.encode(URL1, convert_to_tensor=True)
embeddings2 = model.encode(URL2, convert_to_tensor=True)
from sentence_transformers import util
util.cos_sim(embeddings1, embeddings2)
```