

**دانشگاه صنعتی امیر کبیر**  
**( پلی تکنیک تهران )**

**دانشکده مهندسی کامپیوتر**

**تمرین سوم درس فهم زبان**

**دکتر زینلی**

**غلامرضا دار ۴۰۰۱۳۱۰۱۸**

**بهار ۱۴۰۱**

## فهرست مطالب

۳	.....فاینتیون کردن DistillBERT بر روی PQuAD
۸	.....آموزش DistillBERT از پایه
۱۲	.....فاینتیون کردن دو زبانه

**نکته در مورد این تمرین:** با توجه به اینکه در کد مربوط به این تمرین از، دستوراتی برای متصل شدن به اکانت Huggingface و Google Drive و ... استفاده شده است، این کد قابلیت تبدیل به صورت ماژولار را نداشت و در نتیجه، کد این تمرین به صورت چند نوت بوک که در Google Colab قابل مشاهده و اجرا هستند ارائه می شود. همچنین خروجی های بیشتر در این نوت بوک ها قابل رویت است.

## ۱) فاین تیون کردن DistillBERT بر روی PQuAD

در این سوال قصد داریم یک مدل بر پایه BERT که بر روی زبان فارسی pre-train شده است را بر روی مجموعه داده پرسش و پاسخ PQuAD فاین تیون کنیم.

```
from datasets import load_dataset

dataset = load_dataset("Gholamreza/pquad")
```

برای این کار ابتدا لازم بود مجموعه داده PQuAD را فراخوانی کنیم. به دلیل آسانی کار با کتابخانه Transformers از Huggingface، تصمیم گرفتیم مجموعه داده این سوال را نیز در وبسایت Huggingface آپلود کنیم تا فراخوانی مجموعه داده به شکل زیر انجام شود.

علاوه بر آسانی دریافت مجموعه داده، کلاس‌های مختلف کتابخانه Transformers نیز به خوبی با شیء خروجی این مجموعه داده کار می‌کنند.

در وبسایت Huggingface یک دوره آموزشی در رابطه با Question Answering به کمک لایبری Transformers وجود دارد که در حل این سوال بسیار مفید بود. این دوره از طریق لینک زیر قابل مشاهده است.

<https://huggingface.co/course/chapter7/7?fw=pt>

### V3.0

#### BERT v3.0 Model

- [HooshvareLab/bert-fa-zwnj-base](#)

#### DistilBERT v3.0 Model

- [HooshvareLab/distilbert-fa-zwnj-base](#)

#### ALBERT v3.0 Model

- [HooshvareLab/albert-fa-zwnj-base-v2](#)

#### ROBERTA v3.0 Model

- [HooshvareLab/roberta-fa-zwnj-base](#)

### V2.0

#### ParsBERT v2.0 Model

- [HooshvareLab/bert-fa-base-uncased](#)

اولین چالشی که وجود داشت انتخاب یک مدل بود که بر روی زبان فارسی نیز pre-train شده باشد. با مراجعه به گیت‌هاب Hooshvare متوجه شدیم که تعدادی مدل مختلف در دسترس هستند. از بین این مدل‌ها برای این تمرین مدل DistillBERT را برگزیدیم. دلیل این انتخاب این بود که طبق ادعاهای سازنده‌های DistillBERT اصلی، این مدل از BERT ۶۰٪ سریع‌تر است در حالی که حدود ۹۵٪ کارایی BERT را دارد. همچنین تعداد پارامترهای این مدل حدود ۷۵ میلیون بود که نسبت به ۱۱۰ میلیون پارامتر BERT<sub>base</sub> یا ۳۴۰ میلیون پارامتر BERT<sub>large</sub> مدل بسیار سبک‌تری است که باعث می‌شود در پروسه انجام تمرین بتوانیم با آسانی بیشتری مدل را آموزش دهیم.

ابتدا مجموعه داده PQuAD را لود میکنیم.

```
DatasetDict({
  train: Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 63994
  })
  validation: Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 7976
  })
  test: Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 8002
  })
})
```

این مجموعه داده شامل ۶۳۹۹۴ داده آموزش، ۷۹۷۶ داده ارزیابی و ۸۰۰۲ داده تست است. هر کدام از این داده‌ها در واقع یک سوال هستند و حدوداً به ازای هر ۵ یا ۷ سوال، یک context مشترک وجود دارد. پس از لود کردن مجموعه داده PQuAD، آن را توکنایز می‌کنیم. پارامترهای توکنایزر مشابه تصویر زیر تنظیم شده‌اند.

```
tokenized_example = tokenizer(
    example["question"],
    example["context"],
    max_length=max_length,
    truncation="only_second",
    return_overflowing_tokens=True,
    return_offsets_mapping=True,
    stride=doc_stride
)
```

max\_length=384, stride=128

پس از انجام توکنایزیشن، مدل ذکر شده را می‌سازیم و با کمک این داده آن را به میزان ۳ epoch فاین تیون میکنیم.

```
[ ] model_name = model_checkpoint.split("/")[-1]
args = TrainingArguments(
    f"{model_name}-finetuned-pquad",
    evaluation_strategy = "epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    num_train_epochs=3,
    weight_decay=0.01,
    push_to_hub=True,
)
```

سایر پارامترهای آموزش مانند نرخ یادگیری را در تصویر مشاهده می‌کنید.

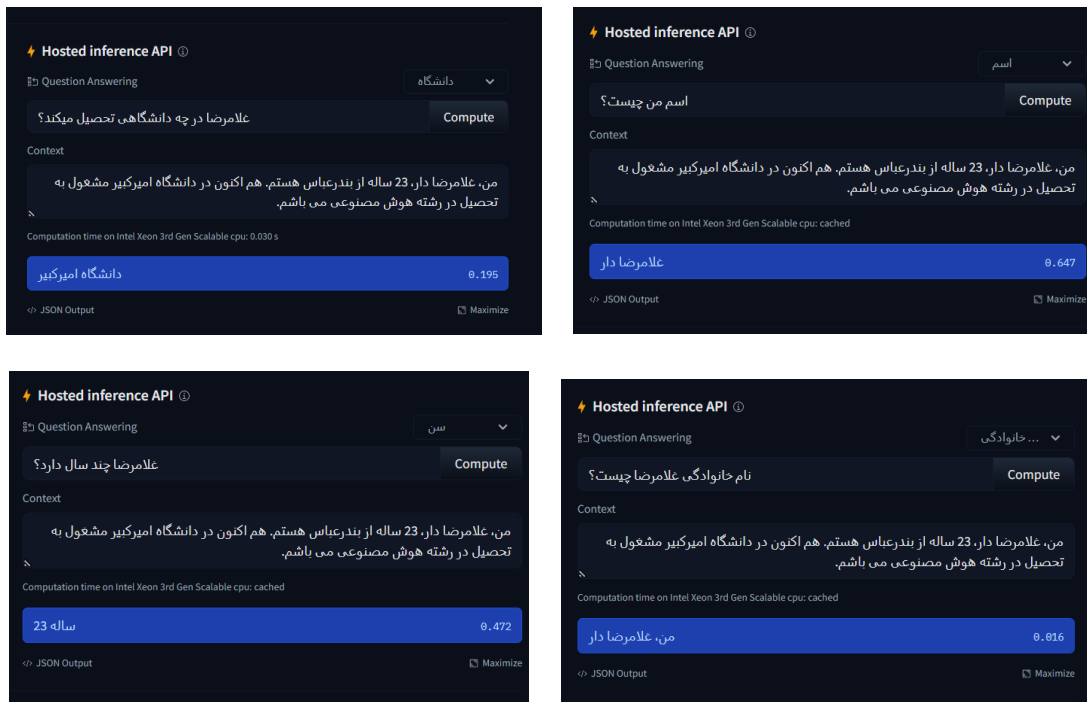
آموزش این مدل به کمک Google Colab در طی مدت ۲ ساعت و نیم انجام شد. نتیجه را در جدول زیر مشاهده می‌کنید.

Model	Parameters	EM	F1	HasAns_EM	HasAns_F1	NoAns_EM/F1
BNA	<a href="#">12M</a>	54.4	71.4	43.9	66.4	87.6
<b>ParsDistillBERT(ours)</b>	75M	66.3	80.2	60.13	78.34	86.2
ParsBERT	110M	68.1	82.0	61.5	79.8	89.0
XLM-RoBERTa	277M	74.8	87.6	69.1	86.0	92.7
Human	-	80.3	88.3	74.9	85.6	96.8

در جلد بالا داده‌های موجود در مقاله مربوط به PQuAD نیز جهت مقایسه آورده شده اند. نتیجه مدل ParsDistillBERT بر روی مجموعه داده تست، را در سطر دوم جدول مشاهده می‌کنید. این مدل از مدل BNA عملکرد بهتری داشت و با اختلاف حدوداً ۰/۲ در معیارهای مختلف، با مدل ParsBERT که مدل بزرگتری است رقابت می‌کند. می‌توان صحت ادعای سازنده‌های DistillBERT مبنی بر ۶۰ درصد سریعتر بودن و دارا بودن ۹۵ درصد کارایی BERT را نشان داد.

با رفتن به لینک زیر می‌توانید عملکرد مدل را در پاسخ‌گویی به سوالات را بررسی کنید.

<https://huggingface.co/Gholamreza/distilbert-fa-zwnj-base-finetuned-pquad>



اگر دقت کنید مدل به سوال نام خانوادگی غلامرضا چیست پاسخ نه چندان مطلوبی داد (کلمات اضافه در پاسخ وجود دارند) اما اگر در متن یک نام خانوادگی رایج‌تر قرار دهیم مدل به درستی نام خانوادگی را استخراج می‌کند.



در ادامه مدل چندزبانه XLM-RoBERTa را نیز فاین تیون کردیم. این مدل کمی بزرگتر از مدل قبلی بود در نتیجه آموزش آن حدود ۴ ساعت طول کشید. نتیجه این مدل به همراه سایر مدل ها را نیز در جدول زیر مشاهده می کنید.

Model	Parameters	EM	F1	HasAns_EM	HasAns_F1	NoAns_EM/F1
BNA	<a href="#">12M</a>	54.4	71.4	43.9	66.4	87.6
ParsDistillBERT(ours)	75M	66.3	80.2	60.13	78.34	86.2
ParsBERT	110M	68.1	82.0	61.5	79.8	89.0
<b>XLM-RoBERTa (ours)<sup>1</sup></b>	277M	74.5	86.7	69.15	85.15	91.8
XLM-RoBERTa	277M	74.8	87.6	69.1	86.0	92.7
Human	-	80.3	88.3	74.9	85.6	96.8

همان طور که انتظار می رفت این مدل هم به دلیل داشتن تعداد پارامتر بیشتر و هم به دلیل نحوه pre-training بهتر، توانست نتیجه بهتری کسب کند. این نتایج با نتایج به دست آمده در مقاله PQuAD بسیار مشابه هستند.

این مدل نیز از طریق لینک زیر قابل دسترس است.

<https://huggingface.co/Gholamreza/xlm-roberta-base-finetuned-pquad>

<sup>1</sup> این مدل فقط یک ایپاک آموزش دید

## ۲) آموزش DistillBERT از پایه

در این بخش از تمرین قصد داریم یک مدل بر پایه BERT را از پایه به کمک داده‌های PQuAD pretrain کنیم. قبل از شروع لازم است کمی در باره مسئله صحبت کنیم. مجموعه PQuAD حدود ۸۰۰۰ متن context متفاوت دارد که هر کدام حدوداً ۳۰۰ توکن طول دارند. این تعداد داده برای آموزش مدلی مانند BERT بسیار کم است. طبق پیشنهاد سوال برای بخش، به جای BERT از DistillBERT که ۴۰ درصد پارامتر کمتر دارد استفاده کردیم اما حتی برای این مدل نیز، چیزی حدود ۱ الی ۲ میلیون توکن آموزش بسیار کم است و حتی در مجموعه داده‌های به نسبت کوچک نیز عددی در مقایسه میلیارد است. انتظار می‌رود این مدل نتواند به خوبی ویژگی‌های زبان فارسی را بیاموزد و در نتیجه در بخش بعد که قرار است این مدل را بر روی دادگان پرسش و پاسخ PQuAD فاین تیون کنیم، دچار مشکلات زیادی شویم.

```
[ ] 1 raw_dataset

Dataset({
  features: ['content'],
  num_rows: 8968
})
```

برای آموزش این مدل، ابتدا مدل را با وزن‌های رندوم init میکنیم، سپس متن‌های context یونیک از مجموعه داده را استخراج می‌کنیم و به عنوان لیستی از متون در می‌آوریم. در ادامه عمل توکنایزیشن را بر روی این مجموعه داده جدید انجام می‌دهیم. پارامترهای توکنایزر را در تصویر زیر مشاهده می‌کنید.

```
outputs = tokenizer(
  element["content"],
  truncation=True,
  max_length=context_length,
  return_overflowing_tokens=True,
  return_length=True,
)
```

میزان max\_length اینجا ۱۲۸ در نظر گرفته شده است. همانطور که ذکر شد اکثر متون در این دیتاست حدود ۳۰۰ توکن دارند، به دلیل در نظر گرفتن مقدار ۱۲۸ برای max\_length، یک متن خاص که طول ۳۰۰ دارد به سه داده با طول‌های ۱۲۸، ۱۲۸، ۴۴ که طبق توصیه‌های موجود در اینترنت نادیده گرفتن آن بخش‌های باقی‌مانده در آخر متون زیاد مشکل ساز نخواهد بود بنابراین ما نیز آن تیکه ۴۴ تایی را حذف میکنیم. در نهایت مجموعه داده ما پس از انجام توکنایزیشن به شکل زیر در خواهد آمد. (۸۰۵۵ بج ۱۲۸ تایی)

```
Dataset({
  features: ['input_ids'],
  num_rows: 8055
})
```



برای آموزش این مدل از پارامترهای زیر استفاده شد.

```
args = TrainingArguments(  
    output_dir=f"{model_checkpoint}-MLM-pquad_2",  
    per_device_train_batch_size=32,  
    per_device_eval_batch_size=32,  
    evaluation_strategy="steps",  
    eval_steps=100,  
    logging_steps=100,  
    gradient_accumulation_steps=8,  
    num_train_epochs=20,  
    weight_decay=0.1,  
    warmup_steps=100,  
    lr_scheduler_type="cosine",  
    learning_rate=8e-4,  
    save_steps=5_000,  
    fp16=True,  
    push_to_hub=True,  
)
```

در جدول زیر روند آموزش این مدل را مشاهده می کنید.

Step	Training Loss
100	6.714000
200	6.421600
300	6.166700
400	5.950200
500	5.657800
600	5.379000

برای بررسی قابلیت های این مدل در مقابل مدل pretrain شده ی ParsBERT از دو لینک زیر استفاده کنید.

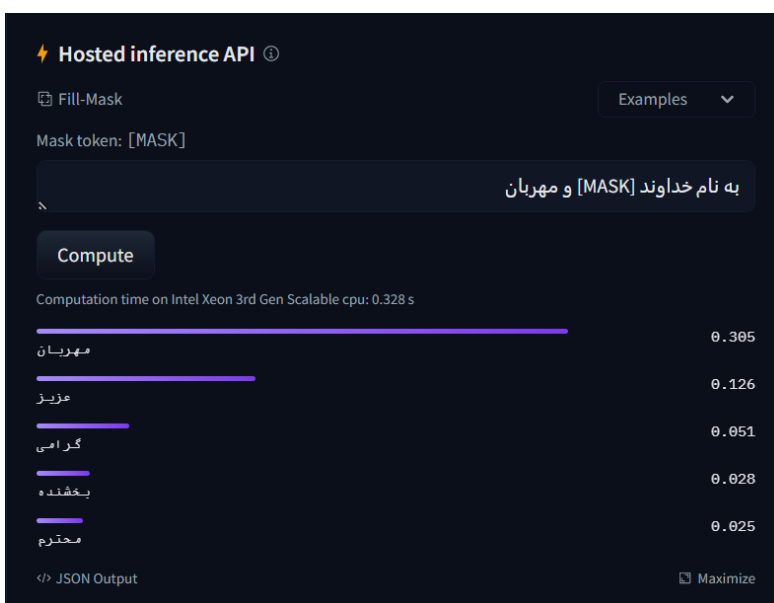
<https://huggingface.co/HooshvareLab/bert-fa-zwnj-base?text=%D8%A8%D9%87+%D9%86%D8%A7%D9%85+%D8%AE%D8%AF%D8%A7%D9%88%D9%86%D8%AF+%5BMASK%5D+%D9%88+%D9%85%D9%87%D8%B1%D8%A8%D8%A7%D9%86>

[https://huggingface.co/Gholamreza/distilbert-fa-zwnj-base-MLM-pquad\\_2?text=%D8%A8%D9%87+%D9%86%D8%A7%D9%85+%D8%AE%D8%AF%D8%A7%D9%88%D9%86%D8%AF+%5BMASK%5D+%D9%88+%D9%85%D9%87%D8%B1%D8%A8%D8%A7%D9%86](https://huggingface.co/Gholamreza/distilbert-fa-zwnj-base-MLM-pquad_2?text=%D8%A8%D9%87+%D9%86%D8%A7%D9%85+%D8%AE%D8%AF%D8%A7%D9%88%D9%86%D8%AF+%5BMASK%5D+%D9%88+%D9%85%D9%87%D8%B1%D8%A8%D8%A7%D9%86)

نمونه Inference به کمک مدل آموزش دیده شده در این تمرین



نمونه Inference به کمک مدل [HooshvareLab/bert-fa-zwnj-base](https://huggingface.co/HooshvareLab/bert-fa-zwnj-base)



نتیجه گیری بخش اول: به احتمال خیلی زیاد می توان با صرف زمان بیشتر و یافتن هایپرپارامترهای مناسب تر، نتیجه این مدل زبانی را بهبود داد اما با توجه به توضیحاتی که در بخش قبل داده شد در نهایت انتظار نمی رود به اندازه مدلی مثل HooshvareLab/bert-fa-zwnj-base عمل کند. به همین دلیل بنده نیز وقت خود را صرف این بخش نکردم و ادامه دادم.

در بخش دوم این سوال، می خواهیم مشابه کاری که در سوال ۱ انجام دادیم را به کمک مدل زبانی pretrain شده خودمان انجام دهیم. تمام بخش های این سوال مشابه سوال ۱ است پس نیاز به توضیح اضافه ای ندارد و صرفا نتایج را نمایش می دهیم.

Model	Parameters	EM	F1	HasAns_EM	HasAns_F1	NoAns_EM/F1
<b>GHDDistillBERT(ours)</b>	75M	24.1	25.9	1.4	3.8	96.2!
BNA	<a href="#">12M</a>	54.4	71.4	43.9	66.4	87.6
ParsDistillBERT(ours)	75M	66.3	80.2	60.13	78.34	86.2
ParsBERT	110M	68.1	82.0	61.5	79.8	89.0
XLM-RoBERTa (ours) <sup>1</sup>	277M	74.5	86.7	69.15	85.15	91.8
XLM-RoBERTa	277M	74.8	87.6	69.1	86.0	92.7
Human	-	80.3	88.3	74.9	85.6	96.8

همان طور که در جدول مشاهده می کنید، این مدل بدترین نتایج را کسب کرد. از آنجایی که بخش fine-tuning این مدل دقیقا شبیه به مدل ParsDistillBERT(ours) که نتایج خوبی بدست آورد است، می دانیم که اینجا صرفا عدم داشتن یک مدل زبانی قدرتمند قدرت مدل را کاهش داده است.

### ۳) فاین تیون کردن دو زبانه

در این سوال قصد داریم یک مدل زبانی چندزبانه مانند XLM-RoBERTa را که بر روی هر دو زبان فارسی و انگلیسی pre-train شده است، به طور همزمان بر روی دو مجموعه داده PQuAD و SQuAD v2 فاین تیون کنیم.

برای این کار می توان کم و بیش از کد مربوط به سوال اول استفاده کرد با این تفاوت که در مرحله آماده سازی مجموعه داده، دو مجموعه داده SQuAD و PQuAD باید با هم ترکیب شوند.

خوشبختانه این دو مجموعه داده از نظر ساختار بسیار به هم شبیه هستند بنابراین کار ترکیب آن ها ساده است. البته این دو مجموعه داده تفاوت هایی دارند که در ادامه با آنها مقابله می کنیم.

اولین تفاوت این است که مجموعه داده SQuAD داده تست ندارد و صرفاً آموزش و اعتبارسنجی است.

```
[9] 1 datasets_squad

DatasetDict({
  train: Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 130319
  })
  validation: Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 11873
  })
})

[10] 1 datasets_pquad

DatasetDict({
  train: Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 63994
  })
  validation: Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 7976
  })
  test: Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 8002
  })
})
```

برای حل این مشکل داده تست مجموعه PQuAD را نیز نادیده می گیریم و مجموعه داده ترکیبی به شکل زیر در خواهد آمد. (در این دیتاست نهایی از نام test استفاده شده که صرفاً یک برچسب است ولی در واقع شامل داده های validation هر دو مجموعه داده است.)

```
DatasetDict({
  train: Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 194313
  })
  test: Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 19849
  })
})
```

تفاوت دومی که این دو مجموعه داده داشتند این بود که ویژگی id در مجموعه داده PQuAD عددی بود اما در SQuAD رشته متنی.

```
[11] 1 datasets_pquad["train"].features
{'id': Value(dtype='float64', id=None),
 'title': Value(dtype='string', id=None),
 'context': Value(dtype='string', id=None),
 'question': Value(dtype='string', id=None),
 'answers': Sequence(feature='text': Value(dtype='string', id=None), 'answer_start': Value(dtype='int32', id=None)), length=-1, id=None)}

[12] 1 datasets_squad["train"].features
{'id': Value(dtype='string', id=None),
 'title': Value(dtype='string', id=None),
 'context': Value(dtype='string', id=None),
 'question': Value(dtype='string', id=None),
 'answers': Sequence(feature='text': Value(dtype='string', id=None), 'answer_start': Value(dtype='int32', id=None)), length=-1, id=None)}
```

بنابراین نیاز بود به شکل زیر دیتاتایپ این ویژگی تغییر کند.

```
1 from datasets import Value, Features, Sequence
2
3 # Match the id datatype
4 datasets_pquad["train"] = datasets_pquad["train"].cast(Features({'id': Value(dtype='string', id=None), 'title': Value(dtype='string', id=None), 'context': Value(dtype='string', id=None), 'question': Value(dtype='string', id=None), 'answers': Sequence(feature='text': Value(dtype='string', id=None), 'answer_start': Value(dtype='int32', id=None))}))
5 datasets_pquad["validation"] = datasets_pquad["validation"].cast(Features({'id': Value(dtype='string', id=None), 'title': Value(dtype='string', id=None), 'context': Value(dtype='string', id=None), 'question': Value(dtype='string', id=None), 'answers': Sequence(feature='text': Value(dtype='string', id=None), 'answer_start': Value(dtype='int32', id=None))}))
6 datasets_pquad["test"] = datasets_pquad["test"].cast(Features({'id': Value(dtype='string', id=None), 'title': Value(dtype='string', id=None), 'context': Value(dtype='string', id=None), 'question': Value(dtype='string', id=None), 'answers': Sequence(feature='text': Value(dtype='string', id=None), 'answer_start': Value(dtype='int32', id=None))}))
```

(تصویر ادامه دارد: همان ویژگی‌هایی که در تصویر بالا خروجی داده شده است را به تابع می‌دهیم.)

مجموعه داده نهایی به شکل زیر در خواهد آمد که شامل سوالات فارسی و انگلیسی می‌باشد. خوب است در این مرحله مجموعه داده را شافل کنیم تا مدل به طور رندوم نمونه‌های فارسی و انگلیسی ببیند.

```
[23] 1 datasets["train"][0]
{'id': '5acfbce377cf76001a685b84',
 'title': 'Umayyad_Caliphate',
 'context': 'The Umayyads have met with a largely negative reception from true caliphate (khilafa). In this respect it is notable that the Umayyad tradition, but rather as khalifat Allah ("deputy of God"). The distinction to share their religious power with, or delegate it to, the emergent class collecting and recording the traditions that form the primary source material such as the histories of Tabari and Baladhuri, that were written in the A',
 'question': 'Who has a positive reception from later Islamic historians?',
 'answers': {'text': [], 'answer_start': []}}

[22] 1 datasets["train"][1]
{'id': '972758',
 'title': 'الکتریسیتہ',
 'context': 'لبرین، مقدار بار ممکن است مثبت یا منفی باشد. طبق قرارداد، باری که به وسیله الکترون، ۱۹-۱۰ کولن را حمل می‌کند نشان می‌دهد. Q است. اندازه بار را با علامت u200c نشان می‌دهد. فرانکلین سرچشمه گرفته اتن تحمل می‌کند؛ بلکه در یادمانده نیز، هر پاندزده باری هم اندازه و مخالف ذره مربوطه u200c نامی، 'شود چیست؟ u200c ها حمل می‌کند طبق قرارداد، باری که به وسیله پروتون، 'answers': {'text': ['متیث'], 'answer_start': [177]}}
```

مدل انتخابی برای این بخش، مدل چندزبانه XLM-RoBERTa بود که آنرا بر روی مجموعه داده ترکیبی جدید فاین تیون کردیم. لازم به ذکر است که در این مرحله گوگل کولب بسیار از دست ما ناراحت بود و نمی گذاشت به GPU دسترسی داشته باشیم. حتی با ساختن اکانت های جدید هم به استفاده ۱ ساعته از GPU محدود بودیم.

```
**** Running Prediction ****
Num examples = 20354
Batch size = 16
[ 3047/12352 1:07:56 < 3:27:36, 0.75 it/s, Epoch 0.25/1]
```

بنابراین این مدل را به میزان حدودا ۲۵ درصد ۱ ایپاک (۳۰۰۰ از ۱۲۰۰۰ ایتريشن) آموزش دادیم که به نتایج زیر رسیدیم. از آنجایی که ادامه این سوال صرفا نیازمند آموزش طولانی مدت تر است، به همین نتیجه بسنده می کنیم. قاعدتا با آموزش بیشتر نتیجه این مدل بهبود پیدا خواهد کرد.

Model	Parameters	EM	F1	HasAns_EM	HasAns_F1	NoAns_EM/F1
<b>XLM-RoBERTa(ours)</b> <b>SQuAD + PQuAD</b>	75M	66.5	73.8	64.5	76.5	69.68

همچنین این مدل نیز در لینک زیر برای ارزیابی قابل دسترس است.

<https://huggingface.co/Gholamreza/xlm-roberta-base-finetuned-pquad-squad>

Context

Computation time on Intel Xeon 3rd Gen Scalable cpu: 0.060 s

Clara

0.546

Context

Computation time on Intel Xeon 3rd Gen Scalable cpu: 0.174 s

غلامرضا

0.670

برخی منابع و نمونه‌های مفید:

ParsBERT : <https://github.com/hooshvare/parsbert>

AUT-NLP : <https://github.com/AUT-NLP/PQuAD>

Pre-training LM course : <https://huggingface.co/course/chapter7/6?fw=pt>

QA fine-tuning course : <https://huggingface.co/course/chapter7/7?fw=pt>

لینک آنلاین نوت‌بوک‌های این تمرین:

### Q1)

Fine-tune DistillBERT:

<https://colab.research.google.com/drive/13oIO5s8qaRHHEs6jcaPMUp0ACz9Rhu7q?usp=sharing>

Fine-tune XLM-RoBERTa:

[https://colab.research.google.com/drive/10\\_oVvazZaO\\_kmb91GNKGUitJdWlrp2wo?usp=sharing](https://colab.research.google.com/drive/10_oVvazZaO_kmb91GNKGUitJdWlrp2wo?usp=sharing)

### Q2)

Pre-train DistillBERT:

[https://colab.research.google.com/drive/1YbZJUAGK1xqBjv1o4Hxz97HmDG4vC\\_e7?usp=sharing](https://colab.research.google.com/drive/1YbZJUAGK1xqBjv1o4Hxz97HmDG4vC_e7?usp=sharing)

Fine-tune the model above :

<https://colab.research.google.com/drive/13wivQBdJkKz3MQGvMPAeYLYc3-e4HqAl?usp=sharing>

### Q3)

Fine-tune on XLM-RoBERTa on en/fa:

[https://colab.research.google.com/drive/1p3E4NatbnL4LmkjvKmsJ\\_IYa-uo7bsaw?usp=sharing](https://colab.research.google.com/drive/1p3E4NatbnL4LmkjvKmsJ_IYa-uo7bsaw?usp=sharing)