

تمرین کامپیوتری چهارم (ثمین مهدی زاده - ۱۴۰۰/۱۰/۸)

1) parsNLU dataset classification

در این سوال سعی شده است تا به کمک روش های contextual word embedding مانند برت تسک Textual entailment انجام شود. برای انجام این تسک در ابتدا از xlm-roberta و سپس از parsbert کمک گرفته شده است و تلاش شده است تا نتایج به دست آمده از هر یک تحلیل شود.

-۱

در ابتدا پس از بررسی داده مشاهده می شود که بعضی از داده ها مقادیر معتبری برای ستون label ندارد و علاوه بر مقادیر c,n,e مقادیری مانند -xx را نیز شامل می شوند. بنابراین لازم است تا این داده ها از دیتاست حذف شوند. مدل های برت از آن جا که بسیار پیچیده و قوی هستند معمولاً احتیاج به پیش پردازش خاصی ندارد و به دلیل pretrain شدن این مدل ها معمولاً مدل به embedding های مناسبی می رسد. علاوه بر این خود دیتاست داده شده هم متن به نسبت مرتب و تمیزی دارد و به نظر نمی رسد انجام پیش پردازش تاثیر زیادی در دقت شبکه داشته باشد. (در قسمت های بعدی مشاهده می شود که تنها با حذف ستون های شامل label های نامعتبر و بدون انجام هیچ پیش پردازش خاصی مدل به دقت مقاله که همان حدود ۵۰ درصد است می رسد.)

```
[ ] df_test['label'].unique()
array(['c', 'n', 'e', '-'], dtype=object)

[ ] df_train['label'].unique()
array(['c', 'n', 'e', 'xx'], dtype=object)

[ ] df_valid['label'].unique()
array(['c', 'e', 'n'], dtype=object)
```

شکل ۱. وجود برجسب های نامعتبر در داده

-۲

در این قسمت برای تولید embedding کلمات موجود در جمله از مدل xlm-roberta استفاده شده است. این مدل یک مدل چند زبانه است که به کمک داده های موجود در چندین زمان مختلف pretrain شده است. برای آموزش این مدل ابتدا لازم است تا ورودی های آن تولید شوند. برای تولید این ورودی ها از تابع encode_plus استفاده شده است. این تابع به این صورت عمل می کند که دو جمله را به عنوان ورودی دریافت می کند و سپس آن ها را به صورت زیر به یکدیگر می چسباند:

[CLS] sent1 [SEP] sent2 [SEP][PAD]...[PAD]

در واقع این تابع پس از توکنایز کردن ورودی ها آن ها را به شکل بالا در می آورد و خروجی هایی مانند input_ids (که بیانگر آیدی توکن های تولید شده) و attention_mask (که با قرار دادن یک در جایگاه هایی که واقعا توکن های مربوط به جمله بودند آن ها را از padding جدا می کند) تولید می کند که به عنوان ورودی به شبکه عصبی داده می شود. خروجی ها نیز از آن جا که از سه کلاس مختلف می باشند به صورت one-hot برای آموزش و ارزیابی مورد استفاده قرار میگیرند. مدل تولید شده در شکل زیر قابل مشاهده است. همچنین در تمامی قسمت های سوال آموزش مدل با lr=3e-5, epoch=10, batch_size=32 صورت گرفته است.

Layer (type)	Output Shape	Param #	Connected to
input_ids (InputLayer)	[(None, 128)]	0	[]
attention_masks (InputLayer)	[(None, 128)]	0	[]
tfxlm_roberta_model (TFXLMRobertaModel)	TFBaseModelOutputWithPoolingAndCrossAttentions(last_hidden_state=(None, 128, 768), pooler_output=(None, 768), past_key_values=None, hidden_states=None, attentions=None, cross_attentions=None)	278043648	['input_ids[0][0]', 'attention_masks[0][0]']
dropout_151 (Dropout)	(None, 768)	0	['tfxlm_roberta_model[0][1]']
dense_6 (Dense)	(None, 1024)	787456	['dropout_151[0][0]']
dense_7 (Dense)	(None, 3)	3075	['dense_6[0][0]']

=====
 Total params: 278,834,179
 Trainable params: 278,834,179
 Non-trainable params: 0

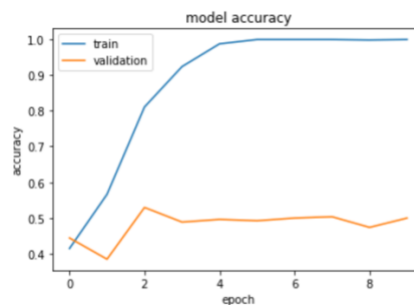
شکل ۲. مدل به کمک xlm-roberta برای entailment

همان طور که مشاهده می شود پس از تولید ورودی ها (input_ids, attention_masks) آن ها را به مدل چند زبانه برت می دهیم تا embedding متناظر با آن ها تولید شود. سپس یک شبکه عصبی که شامل یک لایه dropout و دو لایه dense است ساخته و به کمک ورودی ها مدل را آموزش می دهیم.

```

Epoch 1/10
24/24 [=====] - 40s 1s/step - loss: 1.1011 - accuracy: 0.3714 - val_loss: 1.1233 - val_accuracy: 0.2889
Epoch 2/10
24/24 [=====] - 23s 957ms/step - loss: 1.0972 - accuracy: 0.3886 - val_loss: 1.1029 - val_accuracy: 0.2889
Epoch 3/10
24/24 [=====] - 23s 975ms/step - loss: 1.0913 - accuracy: 0.4151 - val_loss: 1.1349 - val_accuracy: 0.2889
Epoch 4/10
24/24 [=====] - 23s 968ms/step - loss: 1.0915 - accuracy: 0.3674 - val_loss: 1.1720 - val_accuracy: 0.2889
Epoch 5/10
24/24 [=====] - 23s 964ms/step - loss: 1.0845 - accuracy: 0.4019 - val_loss: 1.1304 - val_accuracy: 0.2889
Epoch 6/10
24/24 [=====] - 23s 966ms/step - loss: 1.0618 - accuracy: 0.4204 - val_loss: 1.0876 - val_accuracy: 0.3815
Epoch 7/10
24/24 [=====] - 23s 970ms/step - loss: 1.0313 - accuracy: 0.5053 - val_loss: 1.0958 - val_accuracy: 0.4444
Epoch 8/10
24/24 [=====] - 23s 968ms/step - loss: 0.9352 - accuracy: 0.5557 - val_loss: 1.0301 - val_accuracy: 0.4630
Epoch 9/10
24/24 [=====] - 23s 970ms/step - loss: 0.8255 - accuracy: 0.6552 - val_loss: 1.0918 - val_accuracy: 0.4741
Epoch 10/10
24/24 [=====] - 23s 969ms/step - loss: 0.6058 - accuracy: 0.7851 - val_loss: 1.3640 - val_accuracy: 0.4370

```



شکل ۳. آموزش xlm-roberta برای entailment

مشاهده می شود که پس از ۱۰ epoch مقدار accuracy برای داده های train به ۷۸ درصد و برای داده های validation به ۴۳ درصد رسیده است. برای ارزیابی مدل نیز از داده های test استفاده شده است و نتایج زیر به دست آمده است:

```
Accuracy: 0.4823670053795577
Classification report:

```

			precision	recall	f1-score	support
c	0.59	0.36	0.45	561		
e	0.44	0.75	0.55	610		
n	0.51	0.29	0.37	502		
accuracy			0.48	1673		
macro avg	0.51	0.47	0.46	1673		
weighted avg	0.51	0.48	0.46	1673		

```
F1 (Micro): 0.4823670053795577
F1 (Macro): 0.45828883896178757
F1 (Weighted): 0.4640955349387524
Recall (Micro): 0.4823670053795577
Recall (Macro): 0.46800201083396176
Recall (Weighted): 0.4823670053795577
Precision (Micro): 0.4823670053795577
Precision (Macro): 0.5125694589614822
Precision (Weighted): 0.5104368791379236
```

شکل ۴. نتایج ارزیابی مدل xlm-roberta برای entailment

مشاهده می شود که دقت مدل آموزش داده شده بر روی داده های تست حدود ۴۸ درصد است.

۳-

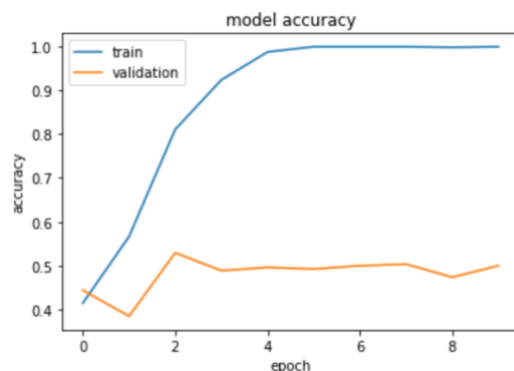
در این قسمت تنها مدل برت استفاده شده متفاوت از قسمت قبل است و شبکه عصبی و ورودی و خروجی های آن تغییری نمی کند. پس از دریافت مدل parsbert آموزش بر روی مدل انجام می شود.

Layer (type)	Output Shape	Param #	Connected to
input_ids (InputLayer)	[(None, 128)]	0	[]
attention_masks (InputLayer)	[(None, 128)]	0	[]
tf_bert_model (TFBertModel)	TFBaseModelOutputWithPoolingAndCrossAttentions(last_hidden_state=(None, 128, 768), pooler_output=(None, 768), past_key_values=None, hidden_states=None, attentions=None, cross_attentions=None)	162841344	['input_ids[0][0]', 'attention_masks[0][0]']
dropout_75 (Dropout)	(None, 768)	0	['tf_bert_model[0][1]']
dense_2 (Dense)	(None, 1024)	787456	['dropout_75[0][0]']
dense_3 (Dense)	(None, 3)	3075	['dense_2[0][0]']

شکل ۵. مدل به کمک parsbert برای entailment

نتایج حاصل از آموزش و ارزیابی مدل در شکل های زیر قابل مشاهده است.

```
Epoch 1/10
24/24 [=====] - 39s 1s/step - loss: 1.1037 - accuracy: 0.4151 - val_loss: 1.0542 - val_accuracy: 0.4444
Epoch 2/10
24/24 [=====] - 22s 922ms/step - loss: 0.9216 - accuracy: 0.5663 - val_loss: 1.1814 - val_accuracy: 0.3852
Epoch 3/10
24/24 [=====] - 22s 913ms/step - loss: 0.6123 - accuracy: 0.8103 - val_loss: 1.0873 - val_accuracy: 0.5296
Epoch 4/10
24/24 [=====] - 22s 911ms/step - loss: 0.2881 - accuracy: 0.9231 - val_loss: 1.2859 - val_accuracy: 0.4889
Epoch 5/10
24/24 [=====] - 22s 915ms/step - loss: 0.0931 - accuracy: 0.9867 - val_loss: 1.7134 - val_accuracy: 0.4963
Epoch 6/10
24/24 [=====] - 22s 917ms/step - loss: 0.0227 - accuracy: 0.9987 - val_loss: 1.8914 - val_accuracy: 0.4926
Epoch 7/10
24/24 [=====] - 22s 916ms/step - loss: 0.0126 - accuracy: 0.9987 - val_loss: 1.9962 - val_accuracy: 0.5000
Epoch 8/10
24/24 [=====] - 22s 914ms/step - loss: 0.0088 - accuracy: 0.9987 - val_loss: 2.0763 - val_accuracy: 0.5037
Epoch 9/10
24/24 [=====] - 22s 913ms/step - loss: 0.0088 - accuracy: 0.9973 - val_loss: 2.3379 - val_accuracy: 0.4741
Epoch 10/10
24/24 [=====] - 22s 911ms/step - loss: 0.0073 - accuracy: 0.9987 - val_loss: 2.2060 - val_accuracy: 0.5000
```



شکل ۶. آموزش مدل *parsbert* برای *entailment*

Accuracy: 0.5218170950388523

Classification report:

			precision	recall	f1-score	support
	c	0.52	0.44	0.48		561
	e	0.52	0.58	0.55		610
	n	0.52	0.54	0.53		502
	accuracy			0.52		1673
	macro avg	0.52	0.52	0.52		1673
	weighted avg	0.52	0.52	0.52		1673

F1 (Micro): 0.5218170950388523

F1 (Macro): 0.5194870845622264

F1 (Weighted): 0.5200366074846379

Recall (Micro): 0.5218170950388523

Recall (Macro): 0.5208152477220852

Recall (Weighted): 0.5218170950388523

Precision (Micro): 0.5218170950388523

Precision (Macro): 0.5216896539647516

Precision (Weighted): 0.5218743765024911

شکل ۷. ارزیابی مدل *parsbert* برای *entailment*

مشاهده می شود که این مدل به دقت ۵۲ درصد بر روی داده های تست رسیده است.

می بینیم که هر دو مدل آموزش داده شده دقت های نزدیک به هم و در حدود مقاله مرجع^۱ دارند. اما شاید کمی نتایج به دست آمده بر روی parsbert بهتر باشد که می تواند به این علت باشد که این مدل تنها برای زبان فارسی آموزش داده شده است و از آن جا که ورودی ها نیز به زبان فارسی بوده اند مدل در یادگیری وزن ها بهتر عمل کرده است اما مدل xlm-roberta نیز به دلیل دیدن زبان های متفاوت و یافتن بازنمایی های مشترک در زبان نیز تا حد خوبی به دقت parsbert رسیده است. نکته دیگری که باید در نظر گرفت این است که xlm-roberta به دلیل پشتیبانی از زبان های زیاد پیچیدگی ها بیشتری دارد و از vocab بیشتری نیز استفاده می کند که می تواند موجب کند بودن آن نسبت به parsbert شود.

Multilingual classification(2)

در این بخش به دسته بندی متون به کمک داده های انگلیسی و فارسی و مدل های تک زبانه و چند زبانه پرداخته شده است و در انتها نیز به تحلیل نتایج به دست آمده پرداخته می شود.

-۱

برای دسته بندی داده های انگلیسی از مدل bert استفاده شده است. نحوه ی ساخت ورودی ها و خروجی ها مانند قسمت قبل و به کمک encode_plus انجام شده است با این تفاوت که در این جا تنها یک متن انگلیسی به عنوان ورودی به تابع داده می شود. شبکه عصبی پیاده سازی شده(لایه bert+یک لایه dropout+دو لایه dense) نیز برای تمام قسمت ها مشابه بوده و تنها مدل های برت آن متفاوت است.

Layer (type)	Output Shape	Param #	Connected to
input_ids (InputLayer)	[(None, 128)]	0	[]
attention_masks (InputLayer)	[(None, 128)]	0	[]
tf_bert_model_1 (TFBertModel)	TFBaseModelOutputWithPoolingAndCrossAttentions(last_hidden_state=(None, 128, 768), pooler_output=(None, 768), past_key_values=None, hidden_states=None, attentions=None, cross_attentions=None)	109482240	['input_ids[0][0]', 'attention_masks[0][0]']
dropout_75 (Dropout)	(None, 768)	0	['tf_bert_model_1[0][1]']
dense_2 (Dense)	(None, 1024)	787456	['dropout_75[0][0]']
dense_3 (Dense)	(None, 3)	3075	['dense_2[0][0]']
=====			
Total params: 110,272,771			
Trainable params: 110,272,771			
Non-trainable params: 0			

شکل ۸. مدل به کمک bert برای دسته بندی متون

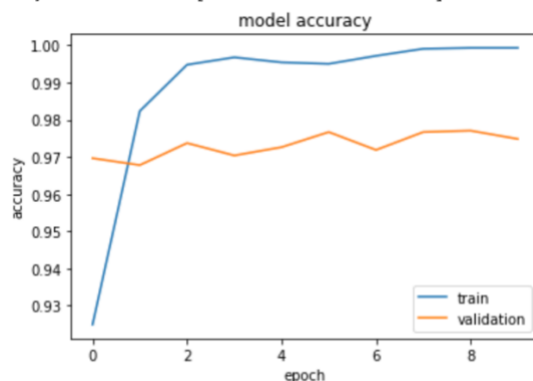
نتایج به دست آمده از آموزش و ارزیابی مدل در زیر آمده است:

¹ <https://arxiv.org/abs/2012.06154>

```

Epoch 1/10
394/394 [=====] - 340s 828ms/step - loss: 0.1943 - accuracy: 0.9249 - val_loss: 0.0868 - val_accuracy: 0.9696
Epoch 2/10
394/394 [=====] - 329s 835ms/step - loss: 0.0528 - accuracy: 0.9823 - val_loss: 0.1154 - val_accuracy: 0.9678
Epoch 3/10
394/394 [=====] - 329s 834ms/step - loss: 0.0168 - accuracy: 0.9948 - val_loss: 0.0994 - val_accuracy: 0.9737
Epoch 4/10
394/394 [=====] - 328s 834ms/step - loss: 0.0100 - accuracy: 0.9967 - val_loss: 0.1242 - val_accuracy: 0.9704
Epoch 5/10
394/394 [=====] - 329s 835ms/step - loss: 0.0130 - accuracy: 0.9954 - val_loss: 0.1142 - val_accuracy: 0.9726
Epoch 6/10
394/394 [=====] - 329s 835ms/step - loss: 0.0146 - accuracy: 0.9950 - val_loss: 0.1105 - val_accuracy: 0.9767
Epoch 7/10
394/394 [=====] - 329s 835ms/step - loss: 0.0087 - accuracy: 0.9971 - val_loss: 0.1099 - val_accuracy: 0.9719
Epoch 8/10
394/394 [=====] - 329s 834ms/step - loss: 0.0030 - accuracy: 0.9990 - val_loss: 0.1120 - val_accuracy: 0.9767
Epoch 9/10
394/394 [=====] - 329s 835ms/step - loss: 0.0038 - accuracy: 0.9993 - val_loss: 0.1018 - val_accuracy: 0.9770
Epoch 10/10
394/394 [=====] - 329s 834ms/step - loss: 0.0032 - accuracy: 0.9993 - val_loss: 0.1475 - val_accuracy: 0.9748

```



شکل ۹. آموزش مدل bert برای دسته بندی متون

Classification report:

	precision	recall	f1-score	support
bible	0.96	0.99	0.97	900
mizan	0.98	0.98	0.98	900
quran	0.99	0.95	0.97	900
accuracy			0.98	2700
macro avg	0.98	0.98	0.98	2700
weighted avg	0.98	0.98	0.98	2700

```

Accuracy: 0.975925925925926
F1 (Micro): 0.975925925925926
F1 (Macro): 0.9759068800565046
F1 (Weighted): 0.9759068800565045
Recall (Micro): 0.975925925925926
Recall (Macro): 0.975925925925926
Recall (Weighted): 0.975925925925926
Precision (Micro): 0.975925925925926
Precision (Macro): 0.9762602252227316
Precision (Weighted): 0.9762602252227316
AUC: 0.9980403292181069

```

شکل ۱۰. ارزیابی مدل bert برای دسته بندی متون

مشاهده می شود که مدل به دقت حدود ۹۷ و مقدار AUC برابر با ۰.۹۹ رسیده است. برای محاسبه AUC نیز از آن جا که چندین کلاس داشتیم از one-vs-rest استفاده شده است به این صورت که به ازای هر کلاس در ابتدا خودش یک طرف و سایر کلاس ها در طرف دیگر قرار میگیرند و AUC برای آن محاسبه می شود و سپس برای محاسبه AUC نهایی میان حالات مختلف میانگین گرفته می شود.

-۲

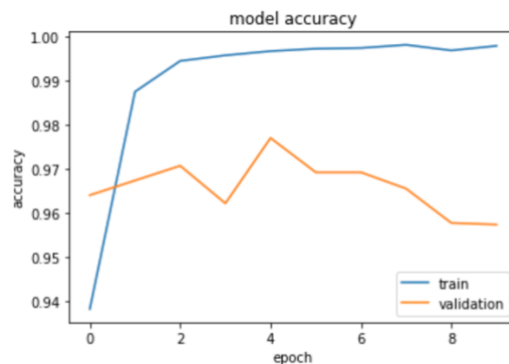
در این بخش تنها داده های آموزشی تغییر می کنند و از داده های فارسی برای دسته بندی استفاده می شود. علاوه بر این برای به دست آوردن embedding ها نیز از مدل parsbert استفاده می شود.

Layer (type)	Output Shape	Param #	Connected to
input_ids (InputLayer)	[(None, 128)]	0	[]
attention_masks (InputLayer)	[(None, 128)]	0	[]
tf_bert_model_2 (TfBertModel)	TfBaseModelOutputWithPoolingAndCrossAttentions(last_hidden_state=(None, 128, 768), pooler_output=(None, 768), past_key_values=None, hidden_states=None, attentions=None, cross_attentions=None)	162841344	['input_ids[0][0]', 'attention_masks[0][0]']
dropout_113 (Dropout)	(None, 768)	0	['tf_bert_model_2[0][1]']
dense_4 (Dense)	(None, 1024)	787456	['dropout_113[0][0]']
dense_5 (Dense)	(None, 3)	3075	['dense_4[0][0]']
=====			
Total params: 163,631,875			
Trainable params: 163,631,875			
Non-trainable params: 0			

شکل ۱.۱. مدل به کمک parsbert برای دسته بندی متون

نتایج به دست آمده از آموزش و ارزیابی مدل فوق در زیر آمده است:

```
Epoch 1/10
394/394 [=====] - 350s 853ms/step - loss: 0.1677 - accuracy: 0.9383 - val_loss: 0.0922 - val_accuracy: 0.9641
Epoch 2/10
394/394 [=====] - 340s 862ms/step - loss: 0.0333 - accuracy: 0.9875 - val_loss: 0.0980 - val_accuracy: 0.9674
Epoch 3/10
394/394 [=====] - 340s 862ms/step - loss: 0.0167 - accuracy: 0.9945 - val_loss: 0.1053 - val_accuracy: 0.9707
Epoch 4/10
394/394 [=====] - 340s 862ms/step - loss: 0.0139 - accuracy: 0.9958 - val_loss: 0.1453 - val_accuracy: 0.9622
Epoch 5/10
394/394 [=====] - 340s 863ms/step - loss: 0.0105 - accuracy: 0.9967 - val_loss: 0.1081 - val_accuracy: 0.9770
Epoch 6/10
394/394 [=====] - 340s 863ms/step - loss: 0.0092 - accuracy: 0.9973 - val_loss: 0.1284 - val_accuracy: 0.9693
Epoch 7/10
394/394 [=====] - 339s 862ms/step - loss: 0.0071 - accuracy: 0.9975 - val_loss: 0.1302 - val_accuracy: 0.9693
Epoch 8/10
394/394 [=====] - 339s 861ms/step - loss: 0.0061 - accuracy: 0.9982 - val_loss: 0.1514 - val_accuracy: 0.9656
Epoch 9/10
394/394 [=====] - 339s 860ms/step - loss: 0.0083 - accuracy: 0.9969 - val_loss: 0.2048 - val_accuracy: 0.9578
Epoch 10/10
394/394 [=====] - 339s 860ms/step - loss: 0.0066 - accuracy: 0.9979 - val_loss: 0.2488 - val_accuracy: 0.9574
```



شکل ۱۱. آموزش مدل *parsbert* برای دسته بندی متون

Classification report:

	precision	recall	f1-score	support
bible	0.93	0.98	0.96	900
mizan	0.98	0.92	0.95	900
quran	0.97	0.97	0.97	900
accuracy			0.96	2700
macro avg	0.96	0.96	0.96	2700
weighted avg	0.96	0.96	0.96	2700

```
Accuracy: 0.9592592592592593
F1 (Micro): 0.9592592592592593
F1 (Macro): 0.9591904391355786
F1 (Weighted): 0.9591904391355787
Recall (Micro): 0.9592592592592593
Recall (Macro): 0.9592592592592593
Recall (Weighted): 0.9592592592592593
Precision (Micro): 0.9592592592592593
Precision (Macro): 0.9601856549747781
Precision (Weighted): 0.960185654974778
AUC: 0.9973330246913581
```

شکل ۱۲. ارزیابی مدل *parsbert* برای دسته بندی متون

همان طور که مشاهده می شود دقت این مدل برای داده های فارسی حدود ۹۶ درصد است که کمی کمتر از حالت قبل است. علت آن می تواند مورفولوژی پیچیده تر زبان فارسی نسبت به انگلیسی و یا وجود داده های بیشتر برای زبان انگلیسی باشد. همچنین اگر به تعداد پارامتر ها نیز توجه کنیم می بینیم که مدل parsbert نسبت به bert پارامتر های بیشتری داشته است.

-۳

این بار از هر دو داده ی فارسی و انگلیسی برای دسته بندی متون استفاده می شود. همچنین از مدل xlm-roberta برای embedding کمک گرفته می شود. برای ترکیب دو داده ی انگلیسی و فارسی مجدداً از encode_plus و به همان صورت که در سوال اول گفته شده بود استفاده می شود تا ترکیب زیر به دست آید:

[CLS] English [SEP] persian [SEP][PAD]...[PAD]

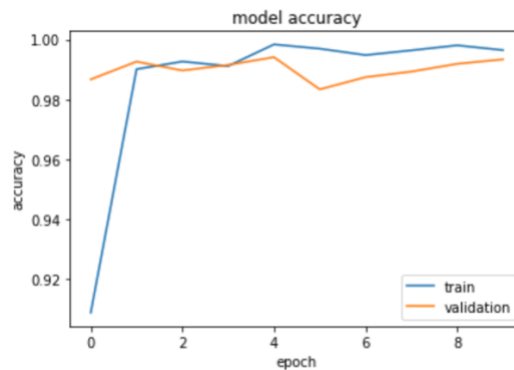
و پس از تولید ورودی های input_ids و attention_mask آموزش بر روی مدل صورت میگیرد.

Layer (type)	Output Shape	Param #	Connected to
input_ids (InputLayer)	[(None, 128)]	0	[]
attention_masks (InputLayer)	[(None, 128)]	0	[]
tfxlm_roberta_model (TFXLMRobertaModel)	TFBaseModelOutputWithPoolingAndCrossAttentions(last_hidden_state=(None, 128, 768), pooler_output=(None, 768), past_key_values=None, hidden_states=None, attentions=None, cross_attentions=None)	278043648	['input_ids[0][0]', 'attention_masks[0][0]']
dropout_151 (Dropout)	(None, 768)	0	['tfxlm_roberta_model[0][1]']
dense_6 (Dense)	(None, 1024)	787456	['dropout_151[0][0]']
dense_7 (Dense)	(None, 3)	3075	['dense_6[0][0]']
=====			
Total params: 278,834,179			
Trainable params: 278,834,179			
Non-trainable params: 0			

شکل ۱۳. مدل به کمک xlm-roberta برای دسته بندی متون

نتایج به دست آمده از آموزش و ارزیابی مدل فوق در زیر آمده است:

```
Epoch 1/10
394/394 [=====] - 382s 933ms/step - loss: 0.2311 - accuracy: 0.9089 - val_loss: 0.0463 - val_accuracy: 0.9867
Epoch 2/10
394/394 [=====] - 365s 926ms/step - loss: 0.0342 - accuracy: 0.9901 - val_loss: 0.0313 - val_accuracy: 0.9926
Epoch 3/10
394/394 [=====] - 365s 926ms/step - loss: 0.0243 - accuracy: 0.9926 - val_loss: 0.0374 - val_accuracy: 0.9896
Epoch 4/10
394/394 [=====] - 365s 926ms/step - loss: 0.0301 - accuracy: 0.9911 - val_loss: 0.0225 - val_accuracy: 0.9915
Epoch 5/10
394/394 [=====] - 365s 926ms/step - loss: 0.0065 - accuracy: 0.9983 - val_loss: 0.0333 - val_accuracy: 0.9941
Epoch 6/10
394/394 [=====] - 365s 927ms/step - loss: 0.0102 - accuracy: 0.9969 - val_loss: 0.0847 - val_accuracy: 0.9833
Epoch 7/10
394/394 [=====] - 365s 927ms/step - loss: 0.0198 - accuracy: 0.9948 - val_loss: 0.0383 - val_accuracy: 0.9874
Epoch 8/10
394/394 [=====] - 365s 926ms/step - loss: 0.0107 - accuracy: 0.9963 - val_loss: 0.0414 - val_accuracy: 0.9893
Epoch 9/10
394/394 [=====] - 365s 927ms/step - loss: 0.0067 - accuracy: 0.9980 - val_loss: 0.0395 - val_accuracy: 0.9919
Epoch 10/10
394/394 [=====] - 365s 927ms/step - loss: 0.0110 - accuracy: 0.9964 - val_loss: 0.0339 - val_accuracy: 0.9933
```



شکل ۱۴. آموزش مدل xlm-roberta برای دسته بندی متون

Classification report:

	precision	recall	f1-score	support
bible	1.00	0.99	0.99	900
mizan	0.99	1.00	0.99	900
quran	0.99	0.99	0.99	900
accuracy			0.99	2700
macro avg	0.99	0.99	0.99	2700
weighted avg	0.99	0.99	0.99	2700

```
Accuracy: 0.9925925925925926
F1 (Micro): 0.9925925925925926
F1 (Macro): 0.9925925647384437
F1 (Weighted): 0.9925925647384437
Recall (Micro): 0.9925925925925926
Recall (Macro): 0.9925925925925926
Recall (Weighted): 0.9925925925925926
Precision (Micro): 0.9925925925925926
Precision (Macro): 0.9926002798629776
Precision (Weighted): 0.9926002798629775
AUC: 0.9995105967078189
```

شکل ۱۵. ارزیابی مدل xlm-roberta برای دسته بندی متون

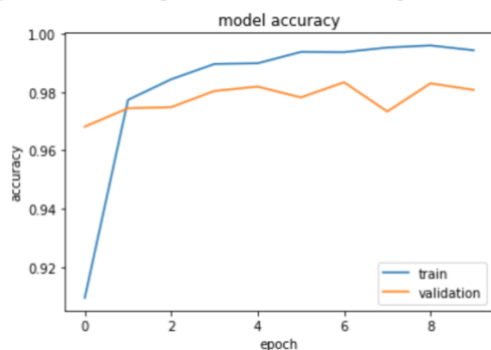
این مدل به نسبت دو مدل قبل به بیشترین میزان دقت و AUC رسیده است. می توان گفت این مدل به علت استفاده از هر دو داده فارسی و انگلیسی یادگیری بیشتری داشته است و به همین علت دقت نهایی آن بیشتر شده است. اگرچه دقت ها در همه موارد بالا و نزدیک به هم بودند که نشان می دهد استفاده از bert شبکه را بسیار قدرتمند می کند.

به طور کلی می توان گفت در این جا بیشترین دقت مربوط به xlm-roberta بوده است و پس از آن به ترتیب مدل های bert و parsebert بوده اند. احتمالاً بهتر عمل کردن bert نسبت به parsbert به این علت است که در parsbert مدل از داده های فارسی استفاده می کند که معمولاً پیچیده تر از زبان انگلیسی است. علاوه براین داده های انگلیسی به نسبت فارسی بسیار بیشتر است که باعث می شود در pretraining مدل انگلیسی بهتر عمل کند و در نتیجه نتایج بهتری بدهد. در این جا به خوبی دیده می شود که استفاده از یک مدل چند زبانه مانند xlm-roberta دقت شبکه را افزایش داده است چرا که این مدل از هر دو داده ی فارسی و انگلیسی استفاده می کند و برای تصمیم گیری از اطلاعات هر دو کمک میگیرد و ممکن است از هر کدام اطلاعاتی را استخراج کند که در متن دیگر نیست و به همین جهت تصمیم گیری بهتری انجام شود البته این در صورتی که است که از داده های هر دو متن با هم استفاده شود برای مثال در سوال اول دیدیم که اگر تنها از داده های فارسی استفاده کنیم مدل parsbert از xlm-roberta بهتر عمل می کند.

Cross-lingual zero-shot transfer learning (3

در این بخش دسته بندی متون به این صورت انجام شده است که برای آموزش از داده های انگلیسی و برای embedding از xlm-roberta استفاده شده است. به هنگام تست نیز داده های فارسی به عنوان ورودی مدل در نظر گرفته شده است. نتایج حاصل از آموزش و ارزیابی این مدل به صورت زیر است:

```
Epoch 1/10
394/394 [=====] - 383s 935ms/step - loss: 0.2358 - accuracy: 0.9095 - val_loss: 0.0911 - val_accuracy: 0.9681
Epoch 2/10
394/394 [=====] - 365s 927ms/step - loss: 0.0710 - accuracy: 0.9773 - val_loss: 0.0739 - val_accuracy: 0.9744
Epoch 3/10
394/394 [=====] - 365s 926ms/step - loss: 0.0454 - accuracy: 0.9844 - val_loss: 0.0898 - val_accuracy: 0.9748
Epoch 4/10
394/394 [=====] - 365s 926ms/step - loss: 0.0311 - accuracy: 0.9896 - val_loss: 0.0965 - val_accuracy: 0.9804
Epoch 5/10
394/394 [=====] - 365s 926ms/step - loss: 0.0301 - accuracy: 0.9898 - val_loss: 0.0794 - val_accuracy: 0.9819
Epoch 6/10
394/394 [=====] - 365s 925ms/step - loss: 0.0194 - accuracy: 0.9937 - val_loss: 0.0970 - val_accuracy: 0.9781
Epoch 7/10
394/394 [=====] - 365s 926ms/step - loss: 0.0180 - accuracy: 0.9937 - val_loss: 0.0803 - val_accuracy: 0.9833
Epoch 8/10
394/394 [=====] - 365s 926ms/step - loss: 0.0174 - accuracy: 0.9952 - val_loss: 0.1235 - val_accuracy: 0.9733
Epoch 9/10
394/394 [=====] - 365s 926ms/step - loss: 0.0117 - accuracy: 0.9960 - val_loss: 0.0958 - val_accuracy: 0.9830
Epoch 10/10
394/394 [=====] - 365s 926ms/step - loss: 0.0190 - accuracy: 0.9943 - val_loss: 0.1051 - val_accuracy: 0.9807
```



شکل ۱۶. آموزش مدل به کمک zero-shot transfer learning برای دسته بندی متون

Classification report:				
	precision	recall	f1-score	support
bible	0.81	0.24	0.37	900
mizan	0.54	1.00	0.70	900
quran	0.78	0.67	0.72	900
accuracy			0.63	2700
macro avg	0.71	0.63	0.60	2700
weighted avg	0.71	0.63	0.60	2700

Accuracy: 0.6344444444444445
 F1 (Micro): 0.6344444444444445
 F1 (Macro): 0.5969503599198224
 F1 (Weighted): 0.5969503599198224
 Recall (Micro): 0.6344444444444445
 Recall (Macro): 0.6344444444444445
 Recall (Weighted): 0.6344444444444445
 Precision (Micro): 0.6344444444444445
 Precision (Macro): 0.7104101946437393
 Precision (Weighted): 0.7104101946437393
 AUC: 0.8305431069958847

شکل ۱۷. ارزیابی مدل به کمک zero-shot transfer learning برای دسته بندی متون

۱-

انتظار می رود که این مدل نیز بتواند تا حدی بر روی داده های فارسی جواب دهد چرا که مدل های چند زبانه سعی می کنند تا یک بازنمایی مشترک برای کلمات در زبان های مختلف پیدا کنند برای مثال در این مدل ها کلمات نزدیک به هم از نظر معنایی در زبان های مختلف در نقاط نزدیک به هم در فضا قرار میگیرند و به همین علت این مدل ها تا حدی می توانند بر روی زبان هایی که در زمان آموزش ندیده اند نیز جواب دهند اگر چه دقت به دست آمده در این حالت قطعا کمتر از زمانی است که داده های train و test در یک زبان باشند.

۲-

پس از اجرای مدل مشاهده می شود که در آخرین epoch دقت بر روی داده های تست برابر با ۹۹ و برای داده های validation برابر با ۹۸ درصد است که نشان می دهد مدل بر روی زبان مشترک خوب عمل می کند. اما هنگامی که داده های تست که به صورت فارسی هستند را به مدل می دهیم دقت حدود ۶۳ درصد داریم که به نسبت حالتی که زبان استفاده شده در آموزش و تست یکسان باشد بسیار پایین تر است. اما نمی توان گفت که مدل این دقت را با تخصیص لیبیل های رندم به دست آورده بلکه قطعا توانسته با پیدا کردن شباهت هایی میان این دو زبان تا حدی دسته بندی را انجام دهد.

۳-

معمولا هنگامی از این روش استفاده می شود که داده های زبان مقصد بسیار کم هستند و نمی توان از آن ها برای آموزش مدل های پیچیده استفاده کرد. در این صورت از داده های زبان های دیگر استفاده کرده به امید آن که مدل با یافتن بازنمایی های مشترک برای زبان ها بتواند بر روی زبان مقصد هم پیش بینی را انجام دهد و به دقت معقولی برسد.