به نام خدا

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده مهندسی برق و کامپیوتر

پردازش زبان‌های طبیعی

تمرین کامپیوتری شماره 4

نام و نام خانوادگی:

امیرحسین دبیری اقدم

شماره دانشجویی:

810197502

خرداد ماه 1401

فهرست

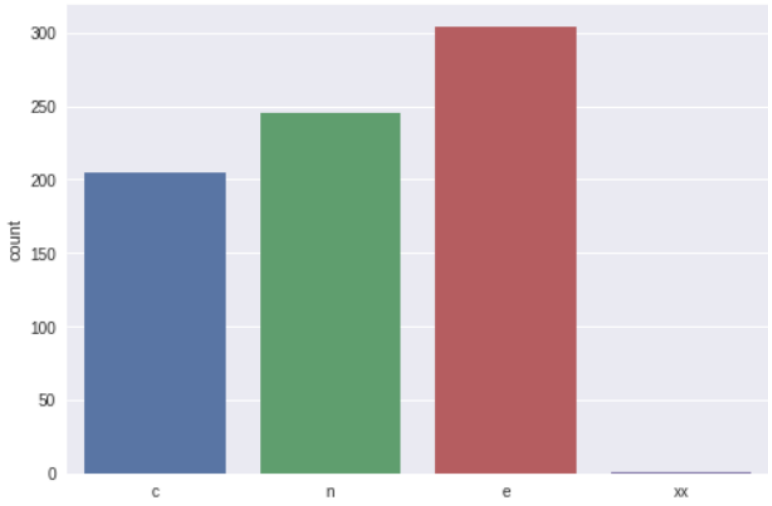
عنوان	شماره صفحه
سوال 1 - ParsiNLU dataset classification	3
سوال 2 - Multilingual classification	9
سوال 3 - Cross-lingual zero-shot transfer learning	14

توضیحات: تمام کدهای مربوط به به هرکدام از پیاده‌سازی‌های ذکر شده در گزارش، در فایل‌های نوت‌بوک ضمیمه شده، آمده است. توضیحات و نتایج نهایی کدها در این گزارش ارائه شده است.

سوال 1 - ParsiNLU dataset classification

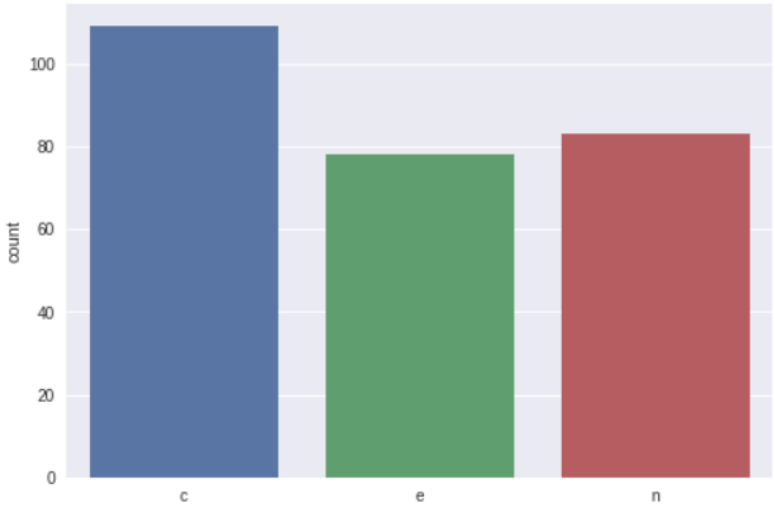
1-در ابتدا توزیع داده‌ها در کلاس‌های مختلف را بررسی می‌کنیم تا مطمئن شویم دیتا balanced است زیرا در غیر اینصورت باید برای هندل کردن این موضوع اقداماتی انجام دهیم زیرا imbalanced بودن داده‌ها بر عملکرد مدل و یادگیری آن اثر نامطلوب دارد. بنابراین توزیع داده‌ها را به صورت bar plot رسم می‌کنیم که در شکل‌های 1 تا 3 قابل مشاهده است.

```
{'c': 0.2718832891246684,
 'e': 0.40318302387267907,
 'n': 0.3249336870026525,
 'xx': 0.001326259946949602}
```

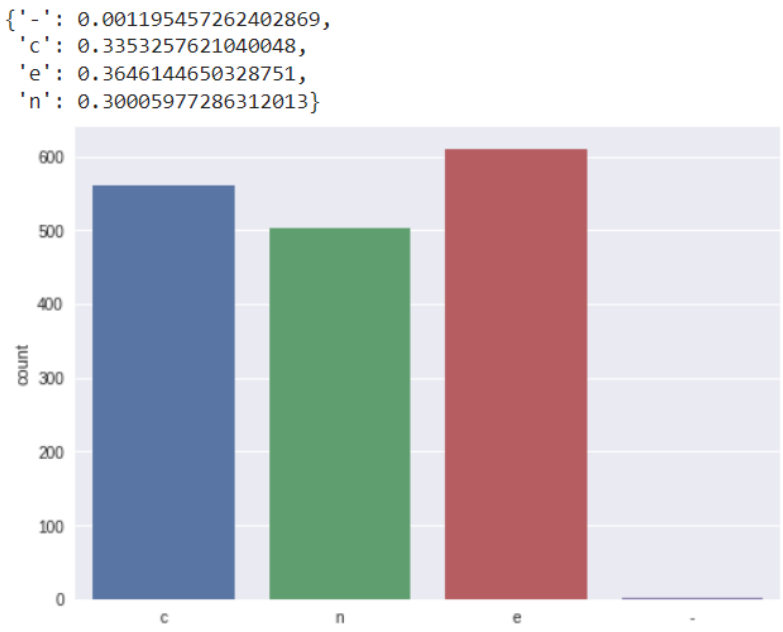


شکل 1 – توزیع داده‌های کلاس‌های مختلف در **train set**

```
{'c': 0.40370370370370373, 'e': 0.28888888888888886, 'n': 0.3074074074074074}
```



شکل 2 - توزیع داده‌های کلاس‌های مختلف در **validation set**



شکل 3 - توزیع داده‌های کلاس‌های مختلف در **test set**

خوشبختانه مشاهده می‌شود که توزیع داده‌ها balanced است و نیازی به عملیات‌هایی نظیر under/over-sampling برای متوازن کردن توزیع داده‌ها نیست. تنها مشکلی که وجود دارد این است که در داده آموزش چند نمونه داده با لیبل xx وجود دارد که با توجه به تعداد کم آن‌ها بهتر است حذف شوند چون احتمالاً مدل نمی‌تواند چیزی درباره آن‌ها یاد بگیرد و داده‌های نویزی هستند بنابراین در مرحله تولید توکن‌های متناظر با هر جمله، این نمونه‌ها را حذف کردم.

همچنین با بررسی گذرای داده‌های آموزش به نظر می‌رسد که داده‌ها به اصطلاح clean هستند و غیر از مورد فوق نیاز به پیش‌پردازش دیگری توسط ما نیست چون از مدل‌های مبتنی بر transformer استفاده می‌کنیم و این مدل‌ها اکثراً بسیار قدرتمند در استخراج ویژگی و کشف الگو حتی از داده‌های خام پیش پردازش نشده هستند. بنابراین تنها کاری که نیاز است، توکنایز کردن جمله‌هاست تا مدل بتواند بر اساس این دیتاست fine-tune شود. برای توکنایز کردن نیز از hugging face tokenizer استفاده می‌کنیم که این توکنایزر متناسب با مدل انتخابی، جملات دیتاست را گرفته و توکن‌هایی که برای آموزش هر مدل نیاز باشد را تولید می‌کند؛ همچین hugging face tokenizer این امکان را می‌دهد که هر دو جمله را به صورت pair در کنار هم با توکن <SEP> قرار دهد و بعد توکنایز کند و خروجی مورد نیاز مدل‌های ترنسفورمر را تولید کند (که در این سوال هم همین کار را می‌کنیم و زوج جملات هر نمونه را به این ترتیب وارد شبکه می‌کنیم.). این توکنایزر همچنین پیش پردازش‌هایی مثل تبدیل حروف به lowercase در صورت نیاز یا تولید token_type_ids برای مدل ParsBert (برای مشخص شدن اینکه هر توکن متعلق به جمله اول pair است یا دوم - مدل XLM-Roberta به این توکن‌ها نیازی ندارد) را هم به صورت خودکار انجام می‌دهد.

به طور کلی این توکنایزر به تنهایی تمام پیش پردازش‌های لازم را انجام می‌دهد و خروجی آن به طور مستقیم قابل استفاده برای آموزش/تست مدل است.

2- با استفاده از مدل XLM-Roberta یک کلسیفایر طراحی می‌کنیم. به این ترتیب که دو جمله که قرار است رابطه بین آن‌ها مشخص شود (Entailment یا Contradiction یا Neutral) را با تگ <SEP> به هم چسبانده و بعد توکنایز کرده و input_ids و attention_mask های متناظر با هر نمونه (که توکنایزر تولید می‌کند) را برای آموزش/ارزیابی مدل استفاده می‌کنیم. در نهایت بردار خروجی مدل متناظر با تگ <CLS> را به یک شبکه عصبی FeedForward دولایه می‌دهیم که activation function آن ReLU است و با اعمال softmax روی خروجی آن، احتمال هر کلاس را محاسبه می‌کنیم.

معماری فوق الذکر را با استفاده از pytorch پیاده‌سازی کرده که در نوت‌بوک متناظر جزئیات پیاده‌سازی آمده است. و با استفاده از train set مدل را با هایپرپارامترهای مشخص شده در جدول زیر (که براساس ارزیابی مدل روی validation set بدست آمده اند) برای این تسک fine-tune می‌کنیم. (stepهای آموزش و ارزیابی در نوت‌بوک مربوطه قابل مشاهده است که برای جلوگیری از شلوغ شدن گزارش صرفا نتیجه نهایی در اینجا ذکر شده است.)

جدول 1- پارامترهای کلسیفایر مبتنی بر XLM-Roberta

HYPER PARAMETER	VALUE
Max sequence length	128
Train batch size	32
Epochs	30
Learning rate	3E-5
Dropout rate	0.3

عملکرد نهایی مدل روی train set به صورت زیر بود:

	precision	recall	f1-score	support
e	0.5728	0.5033	0.5358	610
c	0.4948	0.5882	0.5375	561
n	0.5064	0.4741	0.4897	502
accuracy			0.5230	1673
macro avg	0.5246	0.5219	0.5210	1673
weighted avg	0.5267	0.5230	0.5225	1673

شکل 4 – عملکرد مدل مبتنی بر **XLM-Roberta** روی داده‌های **test** در تسک **Textual entailment** فارسی

مشاهده می‌شود که مدل عملکرد خیلی بالایی ندارد و دقتی در حدود 52.3٪ روی داده test و نیز validation دارد هر چند که روی داده train دقت بالا (حدود 90٪) دارد و اگر تعداد epochها افزایش پیدا می‌کرد حتی دقت روی train افزایش پیدا می‌کرد اما دقت روی validation و test تغییری نمی‌کرد یا بدتر می‌شد که احتمالاً نشانه overfit شدن مدل است البته که شگفتی مدل‌های مبتنی بر transformer که روی داده کمی fine-tune می‌شوند این است که در عین اینکه به نظر می‌رسد روی آن مجموعه داده overfit شده است اما در عین حال دقت نسبتاً قابل قبولی روی داده‌های جدید و دیده نشده دارد که نشان می‌دهد تا حدی مدل توانسته مسئله را به صورت General بیاموزد و اصطلاحاً Generalize شود. البته خود مقاله‌ای که در اینجا ما از دیتاست آن استفاده کردیم هم دقتی در همین حدود، با مدل‌های چندزبانه بر پایه Bert بدست آورده بود که نشان دهنده این است که این تسک (Textual entailment فارسی) پیچیدگی‌های فراوانی دارد و حتی مدل‌های قدرتمندی نظیر XLM-Roberta نیز در حل آن با چالش مواجه هستند.

3 – در همان معماری قسمت قبل این بار به جای XLM-Roberta از مدل ParsBERT استفاده می‌کنیم با استفاده از train set مدل را با هایپرپارامترهای مشخص شده در جدول 2 (که براساس ارزیابی مدل روی validation set بدست آمده اند) برای این تسک fine-tune می‌کنیم.

جدول2 - پارامترهای کلسیفابر مبتنی بر ParsBERT

HYPER PARAMETER	VALUE
Max sequence length	64
Train batch size	32
Epochs	30
Learning rate	2E-5
Dropout rate	0.2

عملکرد نهایی مدل روی train set به صورت زیر بدست آمد:

	precision	recall	f1-score	support
e	0.3641	0.6852	0.4755	610
c	0.3455	0.1016	0.1570	561
n	0.3222	0.2311	0.2691	502
accuracy			0.3533	1673
macro avg	0.3439	0.3393	0.3006	1673
weighted avg	0.3453	0.3533	0.3068	1673

شکل 5 - عملکرد مدل مبتنی بر **Parsbert** روی داده‌های **test** در تسک **Textual entailment** فارسی

مشاهده می‌شود که عملکرد این مدل از قبلی هم بدتر است و عملاً دقت آن اندکی از یک random classifier بیشتر است. مدل با هایپرپارامترهای دیگر هم دقتی در همین حدود می‌گرفت بنابراین ظاهراً این مدل در encode کردن رابطه معنایی دو جمله به صورت کلی ضعیف تر از XLM-Roberta عمل کرده به طوری که ظاهراً روی داده آموزش کاملاً overfit شده و با اینکه دقت آن روی داده‌های train به بیشتر از 99٪ هم می‌رسد اما عملکرد آن روی داده unseen (داده‌های test,validation) بسیار پایین بوده و در حد یک random classifier است! یعنی می‌توان گفت مدل تقریباً Generalize نشده که علت آن، کمبود داده‌های آموزش و/یا عدم تنظیم هایپرپارامترها به صورت مناسب دانست که با توجه به وقت اندک و محدودیت‌های colab بهترین نتیجه‌ای که توانستم بگیرم درحدود همین 35-36٪ بود. البته به ازای برخی هایپرپارامترها به خصوص وقتی تعداد epochها کم بود مدل فقط دو کلاس را خروجی می‌داد (کلاس‌های e,n) اما دقت در این حالت به حدود 38-39٪ هم می‌رسید.

مثلاً با هایپرپارامترهای جدول 3 خروجی شکل 6 حاصل شد.

جدول3 - پارامترهای کلسیفایر مبتنی بر ParsBERT – حالتی که مدل فقط دو کلاس را خروجی می‌داد

HYPER PARAMETER	VALUE
Max sequence length	32
Train batch size	16
Epochs	5
Learning rate	2E-5
Dropout rate	0.0

	precision	recall	f1-score	support
e	0.3703	0.9590	0.5342	610
c	0.0000	0.0000	0.0000	561
n	0.4946	0.0916	0.1546	502
accuracy			0.3772	1673
macro avg	0.2883	0.3502	0.2296	1673
weighted avg	0.2834	0.3772	0.2412	1673

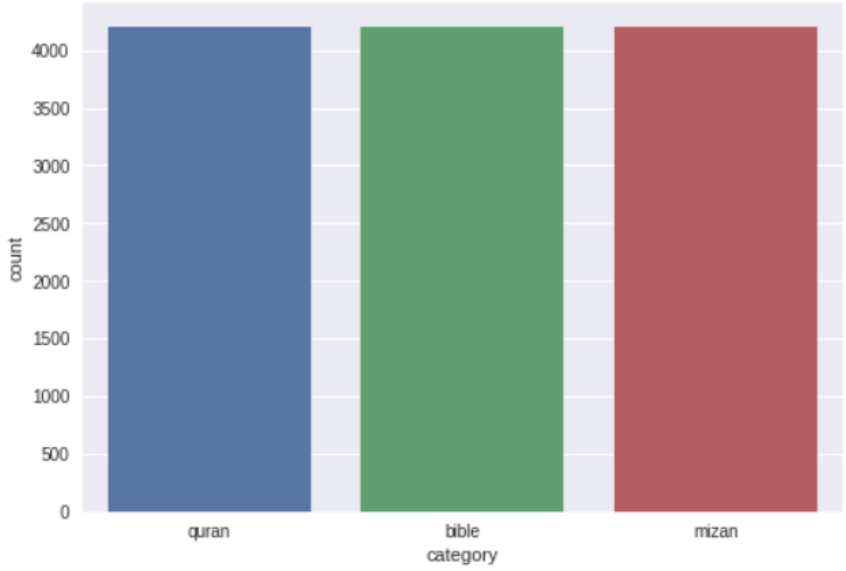
شکل 6 - عملکرد مدل مبتنی بر **Parsbert** روی داده‌های **test** – حالتی که مدل فقط دو کلاس **e,n** را خروجی می‌داد.

در این حالت هم اگرچه دقت اندکی بهتر شده و از random classifier فاصله بیشتری گرفته اما باز هم مشاهده می‌شود که عملکرد آنچنان قابل قبولی نیست و صرفاً دو کلاس از سه کلاس را با دقت نه چندان خوب تا حدی توانسته طبقه‌بندی کند.

سوال 2 - Multilingual classification

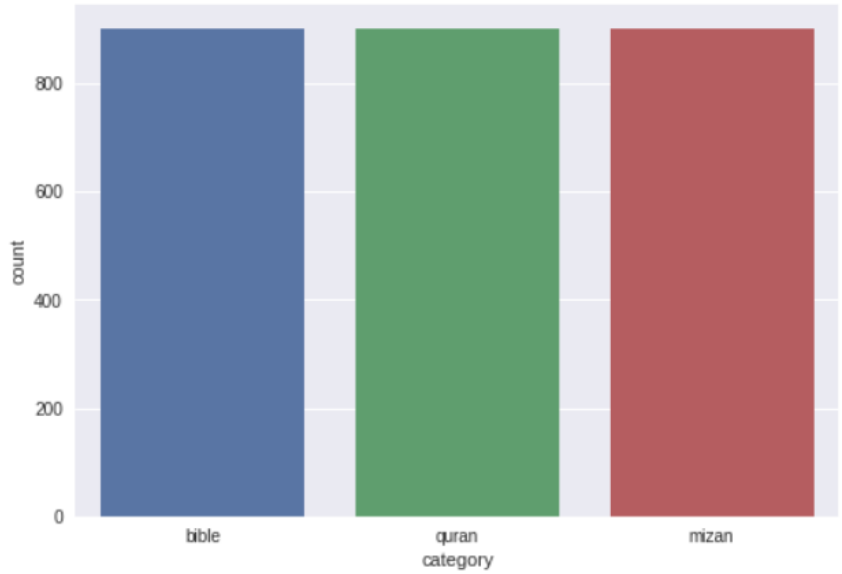
در این سوال هم ابتدا داده‌ها را بررسی می‌کنیم تا ببینیم که آیا نیاز به پیش پردازش خاصی دارد یا خیر بنابراین در ابتدا توزیع نمونه‌ها را در کلاس‌های مختلف رسم می‌کنیم. (شکل‌های 7 الی 9)

```
{'bible': 0.3333333333333333,
 'mizan': 0.3333333333333333,
 'quran': 0.33325396825396825}
```

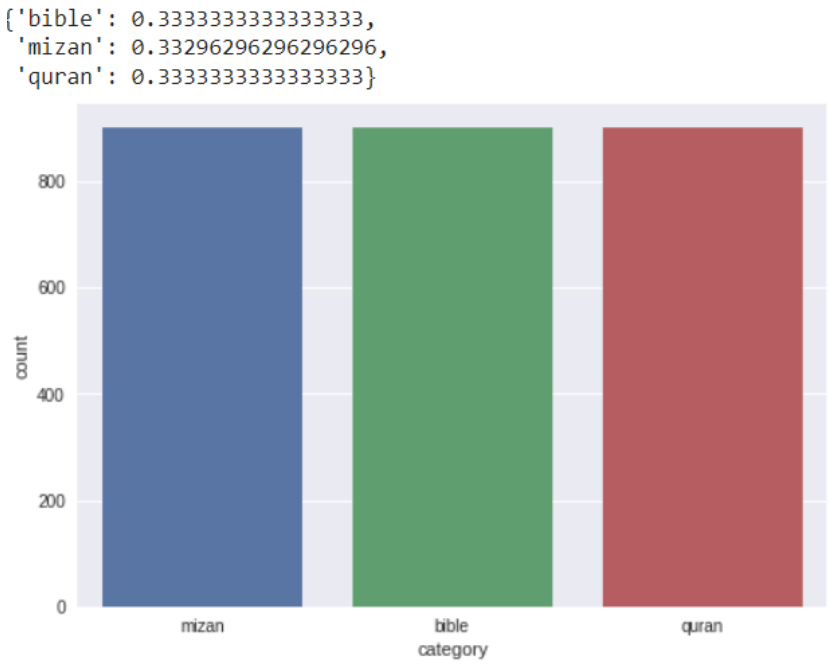


شکل 7 – توزیع نمونه‌های **train set** در کلاس‌های مختلف

```
{'bible': 0.33296296296296296,
 'mizan': 0.3333333333333333,
 'quran': 0.3333333333333333}
```



شکل 8 - توزیع نمونه‌های **validation set** در کلاس‌های مختلف



شکل 9 - توزیع نمونه‌های **test set** در کلاس‌های مختلف

مشاهده می‌شود که توزیع داده‌ها کاملاً متوازن است و نیازی به بالانس کردن آن‌ها نیست. و مشابه سوال قبل با بررسی نمونه‌ها مشاهده می‌شود که به اندازه کافی clean هستند که نیازی به پیش پردازش دیگری ندارد و مشابه قبل کافی است آن را با کمک huggingface tokenizer توکنایز کنیم و...

1-کاملاً مشابه سوال قبل معماری شبکه‌ای که طراحی می‌کنیم به این ترتیب است که یک مدل براساس transformer (در این بخش Bert) جملات ورودی را (که با hugging face tokenizer به صورتی قابل فهم توسط مدل درآمده است) گرفته و representationای که برای جمله ورودی تولید کرده (بردار متناظر با تگ <CLS>) را به یک شبکه FeedForward دولایه (ReLU به عنوان تابع فعالساز و...) می‌دهیم تا عمل classification را برای ما انجام دهد.

پس معماری فوق الذکر را کاملاً مشابه سوال قبل با استفاده از pytorch پیاده‌سازی کرده و با استفاده از ستون مربوط به جمله‌های انگلیسی train set مدل را با هایپرپارامترهای مشخص شده در جدول 4 (که در صورت پروژه ذکر شده بود) برای این تسک fine-tune می‌کنیم. (استپ‌های آموزش و ارزیابی در نوت‌بوک مربوطه قابل مشاهده است که برای جلوگیری از شلوغ شدن گزارش صرفا نتیجه نهایی در اینجا ذکر شده است.)

جدول 4- پارامترهای کلسیفایر مبتنی بر Bert

HYPER PARAMETER	VALUE
Max sequence length	128
Train batch size	32
Epochs	10
Learning rate	3E-5
Dropout rate	0.2

نتیجه عملکرد مدل روی داده test (شامل accuracy,f1-score,AUC و ... که با استفاده از توابع کتابخانه sklearn بدست آمده‌اند) در شکل زیر دیده می‌شود:

	precision	recall	f1-score	support
quran	0.9854	0.9744	0.9799	900
bible	0.9813	0.9900	0.9856	900
mizan	0.9834	0.9856	0.9845	900
accuracy			0.9833	2700
macro avg	0.9833	0.9833	0.9833	2700
weighted avg	0.9833	0.9833	0.9833	2700

AUC = 0.9980

شکل 10 - عملکرد مدل مبتنی بر **bert** روی داده‌های**test** (انگلیسی)

مشاهده می‌شود که شبکه به دقت بالایی روی داده‌های test رسیده که نشان می‌دهد مدل با تنها 10 ایپاک به خوبی Generalize شده زیرا دقت آن روی هر سه مجموعه train,validation,test بالای 98٪ است. همچنین f1-score و AUC یا مساحت زیر نمودار ROC (در واقع میانگین macro این مساحت برای هر کلاس در حالت one-vs-rest) نیز بسیار به یک که حالت ایده‌آل است نزدیک است.

2-همان مدل بخش قبل را استفاده کرده و صرفاً به جای Bert از ParsBert استفاده کرده و نیز از ستون targets که شامل داده‌های به زبان فارسی است برای آموزش مدل استفاده می‌کنیم. (با همان هایپرپارامترهای جدول 4)

نتیجه عملکرد مدل روی داده test در شکل 11 آمده است:

	precision	recall	f1-score	support
quran	0.9680	0.9733	0.9706	900
bible	0.9775	0.9644	0.9709	900
mizan	0.9592	0.9667	0.9629	900
accuracy			0.9681	2700
macro avg	0.9682	0.9681	0.9682	2700
weighted avg	0.9682	0.9681	0.9682	2700

AUC = 0.9963

شکل 11 - عملکرد مدل مبتنی بر **ParsBert** روی داده‌های**test** (فارسی)

باز هم مشاهده می‌شود که شبکه عملکرد خیلی خوبی از خود نشان داده و به خوبی Generalize شده و دقت و ... به طرز قابل قبولی بالا است. البته عملکرد آن نسبت به بخش قبل اندکی افت کرده که قابل انتظار بود زیرا که زبان فارسی زبان پیچیده‌تری است نسبت به انگلیسی و اینکه ParsBert مدلی است براساس Bert که برای فارسی fine-tune شده و طبیعتا انتظار نمی‌رود این مدل روی فارسی دقت یکسانی با Bert (که با حجم خیلی زیادی داده انگلیسی pre-train شده) روی انگلیسی بدهد اما باز هم همانطور که در شکل فوق مشاهده می‌شود دقت خیلی بالایی گرفته و صرفاً به طور میانگین دو درصد از قبلی کمتر است که ممکن است به کیفیت ترجمه و جملات فارسی هم مرتبط باشد این افت عملکرد.

3- همان مدل بخش قبل را استفاده کرده و از XLM-Roberta استفاده کرده و نیز جملات ستون targets که شامل داده‌های به زبان فارسی است را به همراه ستون source که شامل داده‌های انگلیسی است به صورت زوج جمله در کنار هم به عنوان ورودی می‌دهیم. (همانطور که در سوال 1 گفته شد توکنایزر hugging face این قابلیت را دارد که جملات به صورت pair به آن داده شود و این توکنایزر با استفاده از تگ <SEP> آن ها را به در کنار هم قرار داده و حاصل را توکنایز کند و خروجی‌های مناسب را تولید کند) برای آموزش مدل استفاده می‌کنیم. (با همان هایپرپارامترهای جدول 4)

	precision	recall	f1-score	support
quran	0.9911	0.9933	0.9922	900
bible	0.9934	0.9978	0.9956	900
mizan	0.9955	0.9889	0.9922	900
accuracy			0.9933	2700
macro avg	0.9933	0.9933	0.9933	2700
weighted avg	0.9933	0.9933	0.9933	2700

AUC = 0.9997

شکل 12 - عملکرد مدل مبتنی بر **XLM-Roberta** روی داده‌های**test** (فارسی+انگلیسی)

مشاهده می‌شود که عملکردی تقریباً بی نقص دارد مدل و تمامی معیارهای آن یعنی accuracy,f1-score,precision,recall, AUC بیشتر از 0.99 بوده و تقریباً به 1 نزدیک هستند. این موضوع روی داده‌های train,validation هم دیده می‌شد که نشان می‌دهد مدل کاملاً این تسک را یاد گرفته و Generalize شده و نسبت به دو بخش قبلی (استفاده از Bert و ParsBert به ترتیب روی داده‌های انگلیسی و فارسی) عملکرد مدل بر اساس تمام معیارها بهبود پیدا کرده و به حالت ایده آل بسیار نزدیک شده است که نتیجه بسیار خرسند کننده‌ای است.

نتیجه گیری:

به عنوان جمعبندی می‌توان گفت که استفاده از مدل‌های چندزبانی بر روی داده‌های چندزبانی می‌تواند باعث افزایش دقت شود که این موضوع در این سوال بررسی و تایید شد. علت این اتفاق را هم شاید بتوان این موضوع دانست که وقتی از داده‌های چند زبانی استفاده می‌کنیم به نوعی حجم داده‌های در دسترس مدل بیشتر است و در حقیقت الگو و فیچرهای بیشتری در دسترس مدل هست تا با استفاده از آن‌ها و اشتراکاتی که در الگوهای زبانی وجود دارد، یک representation مناسب از هر نمونه ارائه دهد که این بازنمایی هرقدر بهتر باشد عملکرد شبکه FeedForward در کلسیفای کردن هم بیشتر خواهد بود. همچنین می‌دانیم که هر قدر داده آموزشی بیشتر باشد یادگیری و جنرالایز شدن مدل به خصوص مدل‌های بزرگ و پیچیده مبتنی بر ترنسفورمر بیشتر است. پس اگر داده‌های چند زبانه در اختیار داشته باشیم استفاده از مدل‌های چندزبانی روی آن‌ها به احتمال زیاد منجر به دقت بالاتر خواهد شد. حتی در برخی زبان‌ها و بخصوص low-resourceها استفاده از مدل‌های چند زبانه از مدل‌هایی که فقط برای آن زبان خاص fine-tune شده‌اند ممکن است نتیجه بهتری بدهد. (مثلاً مقاله‌ای در سایت medium ادعا کرده که در زبان آلمانی و در برخی تسک‌ها XLM-Roberta Large بهتر از GermanBert عمل می‌کند..)

سوال 3- Cross-lingual zero-shot transfer learning

همان مدل XLM-Roberta بخش آخر سوال دو را استفاده می‌کنیم با این تفاوت که آموزش و ارزیابی آن را صرفاً با داده انگلیسی (ستون source) بوده و در نهایت بدون تغییر وزن‌ها برای تست از داده فارسی (ستون targets) استفاده می‌کنیم.

1- با توجه به عملکرد عالی ای که در بخش آخر سوال قبل از این مدل رو داده‌های چند زبانی دیدم انتظار می‌رود که در این حالت هم performance قابل قبولی از مدل بگیریم زیرا که توکن متناظر با تگ <CLS> که توسط XLM-Roberta تولید می‌شود یک بازنمایی از کلیت جمله براساس محتوا، لحن و... است و شاید این بازنمایی مستقل از زبان باشد. به عبارت دیگر دو جمله که محتوا و لحن و بیان یکسانی داشته باشند مستقل از اینکه با چه زبانی بیان شده‌اند احتمالاً بردار بازنمایی‌هایشان که توسط XLM-Roberta تولید می‌شود به هم نزدیک است (در فضای embedding). بنابراین دیتاست ما که شامل متن‌هایی از کتاب قرآن، کتاب انجیل و متون ادبی است چون لحن و محتوا و بیان متفاوتی دارند مستقل از اینکه مدل با چه زبانی روی این داده‌ها آموزش ببیند، می‌تواند نمونه‌های متناظر با هر کلاس را با دقت نسبتاً خوبی از هم تفکیک کند. پس به صورت خلاصه انتظار داریم عملکرد خوبی را روی داده تست به زبان فارسی شاهد باشیم.

2- با استفاده از همان هایپرپارامترهای قبلی (جدول 4) همانطور که گفته شد که آموزش و ارزیابی آن را صرفاً با داده انگلیسی و تست را با داده فارسی انجام می‌دهیم که نتیجه نهایی در شکل 13 قابل مشاهده است:

	precision	recall	f1-score	support
quran	0.7908	0.7600	0.7751	900
bible	0.8377	0.5389	0.6558	900
mizan	0.6998	0.9767	0.8154	900
accuracy			0.7585	2700
macro avg	0.7761	0.7585	0.7488	2700
weighted avg	0.7761	0.7585	0.7488	2700

AUC = 0.9117

شکل 13- عملکرد مدل مبتنی بر **XLM-Roberta** آموزش دیده روی داده **train** انگلیسی و ارزیابی شده روی داده‌های**test** فارسی

مشاهده می‌کنیم که عملکرد قابل قبولی دارد و انتظارات ما را برآورده کرده است زیرا که دقت حدود 0.91 و AUC در حدود 0.91 دارد که با توجه به اینکه مدل با هیچ نمونه فارسی در هنگام آموزش مواجه نشده بود (به همین دلیل به آن zero-shot می‌گویند)، عملکرد بسیار خوبی از خود نشان داده است. شاید اگر

مقدار اندکی داده فارسی هم برای fine-tune کردن مدل بعد از fine-tune کردن آن روی انگلیسی هم داشتیم، دقت مدل باز هم افزایش می‌یافت. یک روش دیگر که باعث افزایش دقت شد این بود که تعداد epochها را کاهش دهیم مثلاً در شکل زیر performance مدل به ازای همان هایپرپارامترهای قبلی اما فقط به ازای یک ایپاک آمده است.

	precision	recall	f1-score	support
quran	0.8018	0.8000	0.8009	900
bible	0.8183	0.7056	0.7578	900
mizan	0.8402	0.9578	0.8951	900
accuracy			0.8211	2700
macro avg	0.8201	0.8211	0.8179	2700
weighted avg	0.8201	0.8211	0.8179	2700

AUC = 0.9394

شکل 14- عملکرد مدل مبتنی بر **XLM-Roberta** آموزش دیده روی داده **train** انگلیسی و ارزیابی شده روی داده‌های**test** فارسی – تنها یک **epoch** آموزش داده شده

مشاهده می‌شود که حتی با یک ایپاک هم به دقت خوب روی هر سه مجموعه داده train,validation,test (که دوتای اول به زبان انگلیسی و آخری به زبان فارسی است) می‌رسیم و دقت روی داده test فارسی در این حالت حتی بهتر از وقتی شد که آموزش با 10 ایپاک انجام می‌شد. شاید این موضوع را اینچنین بتوان توجیه کرد که وقتی تعداد ایپاک‌ها زیاد است مدل علاوه بر لحن و محتوای کلی جملات سعی می‌کند در داده‌های انگلیسی دقیق‌تر شده و الگوهای دیگری نیز در داده‌های آموزش انگلیسی کشف کند که این الگوها مختص زبان انگلیسی بوده و باعث بهبود عملکرد مدل روی داده‌های انگلیسی می‌شود (شاهد این مدعا افزایش دقت روی داده validation است که تا +98٪ هم می‌رسد). اما این الگوها در جملات به زبان فارسی وجود ندارد یا به هر نحو، طوری است که باعث افت عملکرد مدل روی زبان فارسی می‌شود و حالتی که مدل هنوز بر الگوهای مختص زبان انگلیسی دقیق نشده و صرفاً لحن و محتوا و بیان جمله را در طبقه‌بندی در نظر می‌گیرد (ایپاک اول)، اصطلاحاً عملکرد cross-lingual آن بهتر است.

به طور کلی نتایج بدست آمده در این سوال نشان می‌دهد که همانطور که در قسمت قبل پیش بینی شد، مدل‌های چندزبانی نظیر XLM-Roberta می‌توانند با یادگیری لحن و محتوا و بیان یک جمله (که می‌تواند مستقل از زبان جمله باشد) این تسک را به خوبی انجام دهند و عملکرد قابل قبولی از خود نشان دهند و به طور کلی برای Cross-lingual transfer learning پتانسیل بالایی دارند به خصوص وقتی زبان‌های مبدا و مقصد با هم مشابهت ساختواژی و ساختاری و ... داشته باشند. این پتانسیل بالا نشان

می‌دهد که مدل‌های چندزبانی بازنمایی برداری شبیه به هم از دو جمله با بیان و مفهوم و لحن یکسان اما با دو زبان متفاوت ارائه می‌کنند (در فضای embedding) که بسیار جالب توجه و نشان از قدرت بالای این مدل‌ها در فراگیری زبان‌ها و درک مفهوم و لحن جملات مستقل از زبان است.

3-روش Cross-lingual zero-shot transfer learning بیشتر برای زبان‌های low resource استفاده می‌شود یا به طور کلی برای تسک‌هایی استفاده می‌شود که ما داده آموزش برای زبان X کم داریم یا اصلاً نداریم (zero resource) اما برای زبان Y (یا چندین زبان دیگر به جز X) داده آموزش فراوان داریم؛ در این حالت یکی از مدل‌های چندزبانی را با استفاده از داده‌های زبان Y و... آموزش می‌دهیم (fine-tune می‌کنیم) و بعد، از knowledge کسب شده توسط مدل برای همان تسک (یا تسک مشابه) در زبان X استفاده می‌کنیم (به اصطلاح knowledge را transfer می‌کنیم). به این ترتیب نیاز به داده لیبل خورده در زبان‌های low resource کمتر خواهد بود و حتی می‌توان داده‌ها را با این روش لیبل زد. نکته قابل توجه این است که همانطور که در این سوال دیدیم، این روش کارساز است (حداقل در برخی تسک‌ها) و می‌توان بسیاری از تسک‌های NLP را با این کار برای زبان‌های low resource نیز اپلای کرد و performance قابل قبولی هم از آن گرفت. شاید حتی روزی این مدل‌های چندزبانی آنقدر پیشرفت کنند که دیگر مدل‌های تک زبانه را در تمام یا اغلب تسک‌ها کنار بزنند و از یک مدل چند زبانی آموزش دیده با داده چند زبان محدود برای اغلب تسک‌ها در اکثر زبان‌ها استفاده شود و بهترین عملکرد را هم داشته باشد. (حالت ایده آل Cross-lingual transfer learning).