**باسمه تعالی**

# گزارش تمرین شماره ۵ درس پردازش زبان طبیعی


**استاد درس:** جناب آقای دکتر باباعلی

**نام دانشجو** : ایمان کیانیان

**شماره دانشجویی:** ۶۱۰۳۰۰۲۰۳

## ۱- مقدمه

در این تمرین قصد داریم عمل دسته بندی را روی dataset پرسیکا انجام دهیم. دیتاست پرسیکا، یک دیتاست شامل ۱۱۰۰۰ متن خبری است که از سایت ایسنا جمع آوری شده است. این متون در دسته های کلی مختلف (۱۱ دسته) جمع آوری شده اند که دسته های مربوطه به صورت زیر هستند:

- آموزشی
- تاریخی
- فقه و حقوق
- اجتماعی
- سیاسی
- مذهبی
- اقتصادی
- علمی
- ورزشی
- بهداشتی
- فرهنگی

از هر دسته ۱۰۰۰ خبر وجود دارد. ابتدا روی این اخبار پیش پردازش انجام میدهیم. سپس داده های تست و آموزشی را جدا میکنیم. مدل های مختلف از قبل آموزش داده شده ی Bert را استفاده میکنیم تا بتوانیم عمل text classification را انجام دهیم. در نهایت بهترین نتیجه ای که کسب شد، دقت تقریبا ۹۰ درصد است. در مقایسه با نتیجه تمرین ۴ که ۸۴,۶۳ درصد بود پیشرفت قابل توجهی از نظر دقت داشتیم.

## ۲- پیش پردازش داده ها

ابتدا داده پرسیکا را لود کردیم . پیش پردازش های اولیه نظیر انتقال داده ها به جدول data frame انجام دادیم و اطلاعات کلی دیتاست را بدست آوردیم.

```
count    10999.000000
mean       412.191836
std        590.737729
min          0.000000
25%        140.000000
50%        241.000000
75%        417.500000
max       6332.000000
Name: news_text, dtype: float64
```

همانطور که در شکل مشاهده میشود، ۱۰۹۹۹ خبر داریم که خبر های ما به طور میانگین ۴۱۲ کلمه دارند. ۲۵ درصد خبر ها حداکثر ۱۴۰ کلمه دارند. ۵۰ درصد داده ها حداکثر ۲۴۱ کلمه دارند و ۷۵ درصد اخبار حداکثر ۴۱۷ کلمه دارند. طولانی ترین خبری که در این پیکره متنی وجود دارد ۶۳۳۲ کلمه دارد. به وضوح ما باید تعداد کلمه ورودی به شبکه عصبی BERT را مشخص کنیم و این یکی از پارامتر های ما در آینده خواهد بود. اطلاعات بیشتر درباره هر دسته خبری به صورت زیر است:

```
The number corresponding to the آموزشي is:        0
The number corresponding to the اجتماعي is:       1
The number corresponding to the اقتصادي is:       2
The number corresponding to the بهداشتي is:       3
The number corresponding to the تاريخي is:        4
The number corresponding to the سياسي is:         5
The number corresponding to the علمي is:          6
The number corresponding to the فرهنگي is:        7
The number corresponding to the فقه و حقوق is:    8
The number corresponding to the مذهبي is:         9
The number corresponding to the ورزشي is:         10
```

همانطور که واضح است برای این پروژه از label encoding استفاده شده است که شماره هایی که از این به بعد ذکر میشوند معادل با کلاس های بالا هستند. مثلا عدد ۴ معادل کلاس تاریخی است و عدد ۹ معادل با کلاس مذهبی.

```
Count of news in subject 1000 = اقتصادي
Count of news in subject 1000 = فرهنگي
Count of news in subject 1000 = علمي
Count of news in subject 1000 = سياسي
Count of news in subject 1000 = فقه و حقوق
Count of news in subject 1000 = ورزشي
Count of news in subject 1000 = اجتماعي
Count of news in subject 1000 = مذهبي
Count of news in subject 1000 = آموزشي
Count of news in subject 999 = تاريخي
Count of news in subject 1000 = بهداشتي
```

که نشان میدهد در این پیکره متنی برای هر خبر به تعداد یکسانی (۱۰۰۰) خبر داریم. بنابراین تعداد اخبار balance هستند.

در قسمت پیش پردازش داده ها ابتدا stop word ها را از این پیکره حذف میکنیم. این کار را با استفاده از لیستی از کلمات ایست که در فایل stop.txt همراه با تمرین آمده است انجام میدهیم. سپس علائم نگارشی مثل "." یا "(" و ... را حذف میکنیم چون فاقد ارزش معنایی هستند و تعداد تکرار آنها مانند stop word ها بسیار زیاد است. لیست کامل علائم نگارشی حذف شده به صورت زیر است:

```
['!','»','،','؟','؛','"','#','(',')','*',',','-','.','/',':','[',']','«','…']
```

با استفاده از متد word_tokenize پکیج Hazm کلمات هر خبر را جدا کردیم و کمی نرمالش کردیم و سپس دوباره جملات اخبار را بعد از پیش پردازش به هم متصل کردیم. داده ها را به ۲ بخش آموزش و تست با نسبت ۸۰ به ۲۰ تقسیم کردیم. سپس ۸۰ درصد آموزشی را با نسبت ۹۰ به ۱۰ بین داده های آموزشی و ارزیابی تقسیم کردیم. سپس وارد مرحله بعد شدیم که استفاده از شبکه های عصبی پیش پردازش شده BERT است.

## ۳- استفاده از شبکه های عصبی پیش پردازش شده BERT

### ۳-۱. استفاده از ParsBERT برای عمل text classification با ماکزیمم طول ۵۱۲:

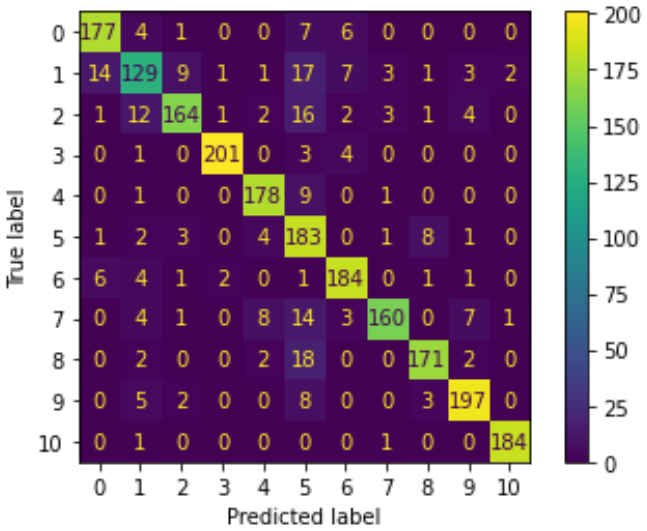
در این قسمت از یک مدل ParsBERT که در سایت Hugging Face موجود است استفاده کردیم (لینک). ماکزیمم طول هر جمله که همان ورودی شبکه BERT ما است برابر ۵۱۲ قرار دادیم ( با توجه به میانگین تعداد کلمات در هر جمله). بقیه پارامتر های fine tuning این مدل به صورت زیر است:

```
output_dir='/content/gdrive/MyDrive/NLP - HW5/results',          # output directory
num_train_epochs=4,              # total number of training epochs
per_device_train_batch_size=8,  # batch size per device during training
per_device_eval_batch_size=20,   # batch size for evaluation
warmup_steps=200,                # number of warmup steps for learning rate scheduler
weight_decay=0.01,               # strength of weight decay
logging_dir='/content/gdrive/MyDrive/NLP - HW5/logs',            # directory for storing logs
load_best_model_at_end=True,     # load the best model when finished training (default metric is loss)
# but you can specify `metric_for_best_model` argument to change to accuracy or other metric
logging_steps=1000,               # log & save weights each logging_steps
save_steps=1000,
evaluation_strategy="steps",     # evaluate each `logging_steps`
```

به دلیل محدودیت منابع پردازشی از ۳ epoch برای fine tuning استفاده کردیم.دقت در حین فرآیند آموزش به صورت زیر است:

| Step | Training Loss | Validation Loss | Accuracy |
|---|---|---|---|
| 1000 | 0.761300 | 0.531617 | 0.873998 |
| 2000 | 0.347100 | 0.528367 | 0.893471 |
| 3000 | 0.187000 | 0.556513 | 0.900344 |

همچنین confusion matrix برای داده های تست به صورت زیر است:




مشاهده میشود دقت برای کلاس ۱ که همان  فرهنگی باشد کمی پایین است. تعداد اخباری که باید فرهنگی پیش بینی میشد ولی نشده است تقریبا بالاست. دقت برای داده های تست در این مرحله برابر ***0.8831882730187814*** است. معیار های دیگر دقت برای داده های تست با جزئیات به شرح زیر است:

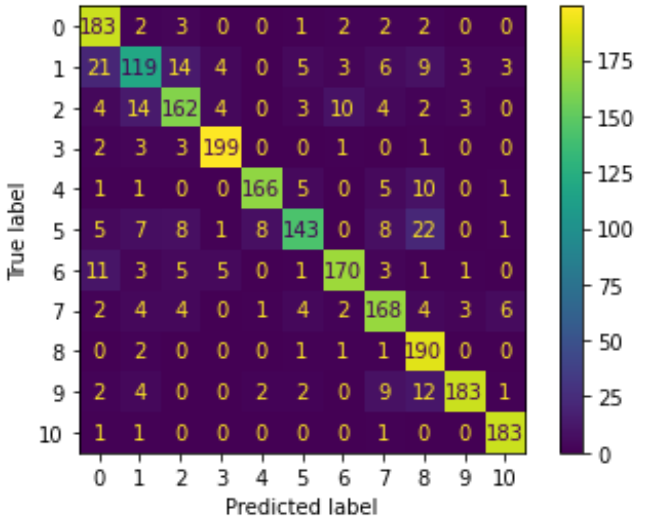
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.89 | 0.91 | 0.90 | 195 |
| 1 | 0.78 | 0.69 | 0.73 | 187 |
| 2 | 0.91 | 0.80 | 0.85 | 206 |
| 3 | 0.98 | 0.96 | 0.97 | 209 |
| 4 | 0.91 | 0.94 | 0.93 | 189 |
| 5 | 0.66 | 0.90 | 0.76 | 203 |
| 6 | 0.89 | 0.92 | 0.91 | 200 |
| 7 | 0.95 | 0.81 | 0.87 | 198 |
| 8 | 0.92 | 0.88 | 0.90 | 195 |
| 9 | 0.92 | 0.92 | 0.92 | 215 |
| 10 | 0.98 | 0.99 | 0.99 | 186 |
| accuracy | | | 0.88 | 2183 |
| macro avg | 0.89 | 0.88 | 0.88 | 2183 |
| weighted avg | 0.89 | 0.88 | 0.88 | 2183 |

### ۲-۳. استفاده از ParsBERT بدون انجام پیش پردازش های اولیه (Hazm و Stemmer و ...):

در این قسمت از همان مدل ParsBERT قسمت قبل استفاده کردیم فقط پیش پردازش های قبلی انجام نشده است. دقت حین فرآیند آموزش:

| Step | Training Loss | Validation Loss | Accuracy |
|---|---|---|---|
| 1000 | 0.828200 | 0.522115 | 0.865979 |
| 2000 | 0.390700 | 0.522856 | 0.879725 |
| 3000 | 0.216700 | 0.550005 | 0.902635 |

ماتریس آشفتگی برای داده های تست به صورت زیر است:




همچنین دقت برای داده های تست در این مورد برابر ***0.8547869903802107*** درصد است که از مورد قبلی ضعیفتر است. جزئیات بیشتر درباره دقت روی داده تست به صورت زیر است:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.79 | 0.94 | 0.86 | 195 |
| 1 | 0.74 | 0.64 | 0.69 | 187 |
| 2 | 0.81 | 0.79 | 0.80 | 206 |
| 3 | 0.93 | 0.95 | 0.94 | 209 |
| 4 | 0.94 | 0.88 | 0.91 | 189 |
| 5 | 0.87 | 0.70 | 0.78 | 203 |
| 6 | 0.90 | 0.85 | 0.87 | 200 |
| 7 | 0.81 | 0.85 | 0.83 | 198 |
| 8 | 0.75 | 0.97 | 0.85 | 195 |
| 9 | 0.95 | 0.85 | 0.90 | 215 |
| 10 | 0.94 | 0.98 | 0.96 | 186 |
| accuracy | | | 0.85 | 2183 |
| macro avg | 0.86 | 0.85 | 0.85 | 2183 |
| weighted avg | 0.86 | 0.85 | 0.85 | 2183 |

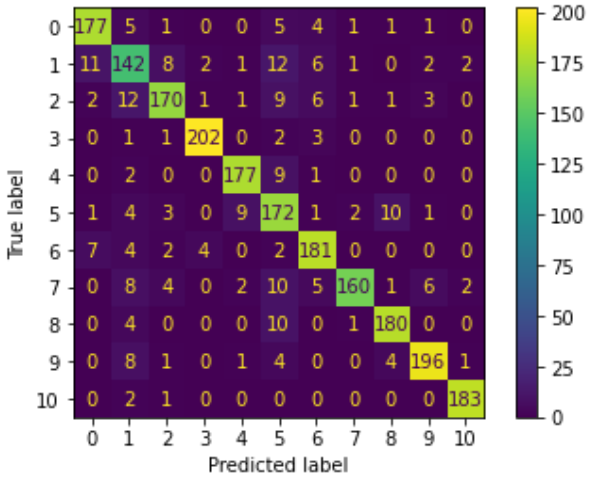
### ۳-۳. استفاده از ParsBERT قسمت اول با max size برابر ۲۵۶:

اینبار میخواهیم اثر کم کردن max size را روی این مدل ببینیم. بنابراین مدل بهتر که مدل ۱-۳ بود را انتخاب میکنیم. پیش پردازش ها همچنان انجام میشوند اما به جای اینکه ماکزیمم طول ۵۱۲ باشد این مقدار را به ۲۵۶ کاهش دادیم. دقت در این مورد در حین آموزش به شکل زیر بود:

| Step | Training Loss | Validation Loss | Accuracy |
|---|---|---|---|
| 1000 | 0.753400 | 0.535330 | 0.877434 |
| 2000 | 0.346900 | 0.463401 | 0.902635 |
| 3000 | 0.193700 | 0.524022 | 0.907216 |

با مشاهده این جدول میتوان دریافت که کمی دچار overfit جزئی میشویم چون loss روی داده های آموزشی به شدت پایین می آید ولی روی داده های validation از جایی به بعد این loss تقریبا ثابت مانده است.( این موضوع در بخش های ۲-۳ و ۱-۳ نیز صادق است)

همچنین ماتریس آشفتگی برای داده های تست در این حالت به صورت زیر است:




دقت برای داده های آموزشی برابر ***0.8886852954649564*** است که از دو حالت قبلی بالاتر است. پس انتخاب این مدل معقول به نظر میرسد چون دقت را برای داده های تست بالاتر برده است. این احتمالا به این دلیل است که با کم کردن سایز جملات ورودی احتمالا یک سری کلمات که مفید نیستند از جملات طولانی حذف شده اند. این باعث شده است که دقت بالاتر باشد. همچنین آموزش مدل در این حالت بسیار ساده تر است چون پارامتر های مدل به شدت کمتر است. همچنین دقت برای داده های تست به صورت جزئی تر برای هر کلاس به صورت زیر است:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.89 | 0.91 | 0.90 | 195 |
| 1 | 0.74 | 0.76 | 0.75 | 187 |
| 2 | 0.89 | 0.83 | 0.86 | 206 |
| 3 | 0.97 | 0.97 | 0.97 | 209 |
| 4 | 0.93 | 0.94 | 0.93 | 189 |
| 5 | 0.73 | 0.85 | 0.79 | 203 |
| 6 | 0.87 | 0.91 | 0.89 | 200 |
| 7 | 0.96 | 0.81 | 0.88 | 198 |
| 8 | 0.91 | 0.92 | 0.92 | 195 |
| 9 | 0.94 | 0.91 | 0.92 | 215 |
| 10 | 0.97 | 0.98 | 0.98 | 186 |
| accuracy | | | 0.89 | 2183 |
| macro avg | 0.89 | 0.89 | 0.89 | 2183 |
| weighted avg | 0.89 | 0.89 | 0.89 | 2183 |

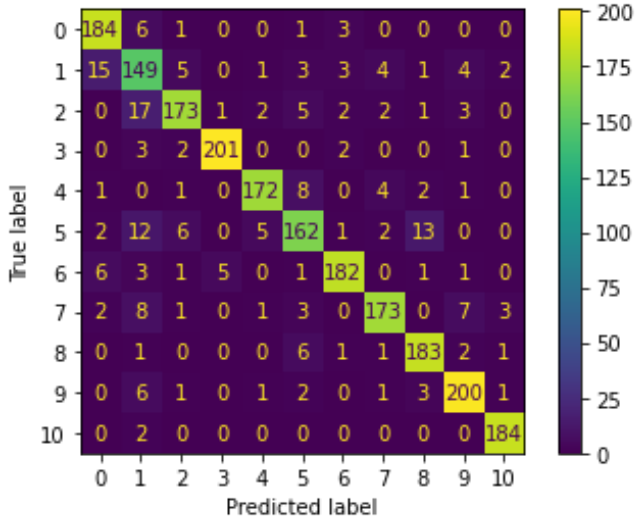
که این نشان میدهد دقت بالاتر است ( کلاس ۱ را در حالت های ۱–۳ و ۲–۳ مقایسه کنید).

### ۴–۳. استفاده از ParsBERT قسمت قبل (طول ۲۵۶) و زیاد کردن weight decay برای جلوگیری از overfitting:

در این قسمت به امید اینکه overfitting کاهش پیدا کند و دقت برای داده های تست کمی افزایش داشته باشد weight decay را از مقدار ۰,۰۱ که در آموزش تمامی ۳ مدل قبلی وجود داشت به مقدار ۰,۱ افزایش دادیم. یعنی این مقدار را ۱۰ برابر کردیم. نتیجه اینکار در قسمت آموزش به صورت زیر است:

| Step | Training Loss | Validation Loss | Accuracy |
|---|---|---|---|
| 1000 | 0.962700 | 0.553823 | 0.871707 |
| 2000 | 0.422400 | 0.546820 | 0.868270 |
| 3000 | 0.207600 | 0.595782 | 0.895762 |

بر اساس این آمار اینکار شاید کمی جلوی overfitting را بگیرد اما احتمالا نتوانسته دقت را افزایش دهد. ماتریس آشفتگی برای داده های تست به صورت زیر است:



دقت برای داده های تست ***0.8992212551534585*** است که یعنی این از تمام ۳ مدلی که قبلا ساختیم دقت بالاتری دارد . پس افزایش weight decay اثر مطلوبی گذاشته است.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.88 | 0.94 | 0.91 | 195 |
| 1 | 0.72 | 0.80 | 0.76 | 187 |
| 2 | 0.91 | 0.84 | 0.87 | 206 |
| 3 | 0.97 | 0.96 | 0.97 | 209 |
| 4 | 0.95 | 0.91 | 0.93 | 189 |
| 5 | 0.85 | 0.80 | 0.82 | 203 |
| 6 | 0.94 | 0.91 | 0.92 | 200 |
| 7 | 0.93 | 0.87 | 0.90 | 198 |
| 8 | 0.90 | 0.94 | 0.92 | 195 |
| 9 | 0.91 | 0.93 | 0.92 | 215 |
| 10 | 0.96 | 0.99 | 0.98 | 186 |
| accuracy | | | 0.90 | 2183 |
| macro avg | 0.90 | 0.90 | 0.90 | 2183 |
| weighted avg | 0.90 | 0.90 | 0.90 | 2183 |

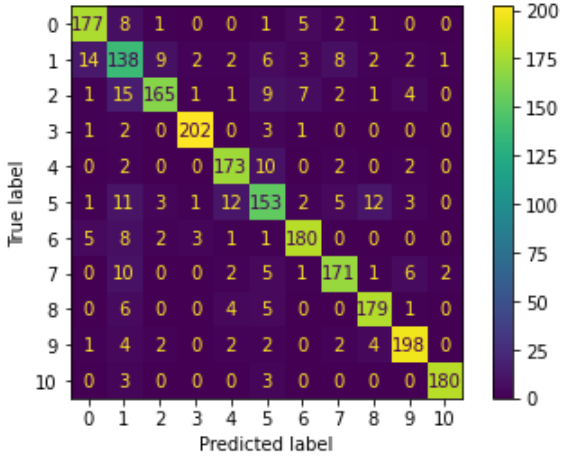
واضح است که recall در کلاس ۱ بسیار بالاتر از حالت های قبلی است. این نتیجه، نتیجه ی خوبی است.

## ۵-۳. استفاده از Multilingual BERT برای عمل Text Classification:

در این قسمت از مدل چند زبانه از پیش آموزش داده شده BERT استفاده کردیم (لینک). این مدل هم نزدیک به مدل های قبل دقت خوبی ارائه میکند. در این مدل از ماکزیمم ۲۵۶ کلمه استفاده کردیم. پارامتر های مدل برای آموزش به صورت زیر است:

```
training_args = TrainingArguments(
    output_dir='/content/gdrive/MyDrive/NLP - HW5/results',          # output directory
    num_train_epochs=5,              # total number of training epochs
    per_device_train_batch_size=8,  # batch size per device during training
    per_device_eval_batch_size=20,   # batch size for evaluation
    warmup_steps=1000,                # number of warmup steps for learning rate scheduler
    weight_decay=0.01,               # strength of weight decay
    logging_dir='/content/gdrive/MyDrive/NLP - HW5/logs',            # directory for storing logs
    #load_best_model_at_end=True,     # load the best model when finished training (default metric is loss)
    # but you can specify `metric_for_best_model` argument to change to accuracy or other metric
    logging_steps=5000,               # log & save weights each logging_steps
    save_steps=5000,
    evaluation_strategy="steps",     # evaluate each `logging_steps`
)
```

ماتریس آشفتگی برای داده های تست در این مدل به صورت زیر است:




همچنین دقت برای داده های تست ***0.8776912505726066*** است که دقت قابل قبولی است به نسبت اینکه مدل ما چند زبانه است. معیار های دیگر برای ارزیابی مدل برای داده های تست به صورت زیر است:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.89 | 0.91 | 0.90 | 195 |
| 1 | 0.67 | 0.74 | 0.70 | 187 |
| 2 | 0.91 | 0.80 | 0.85 | 206 |
| 3 | 0.97 | 0.97 | 0.97 | 209 |
| 4 | 0.88 | 0.92 | 0.90 | 189 |
| 5 | 0.77 | 0.75 | 0.76 | 203 |
| 6 | 0.90 | 0.90 | 0.90 | 200 |
| 7 | 0.89 | 0.86 | 0.88 | 198 |
| 8 | 0.90 | 0.92 | 0.91 | 195 |
| 9 | 0.92 | 0.92 | 0.92 | 215 |
| 10 | 0.98 | 0.97 | 0.98 | 186 |
| accuracy | | | 0.88 | 2183 |
| macro avg | 0.88 | 0.88 | 0.88 | 2183 |
| weighted avg | 0.88 | 0.88 | 0.88 | 2183 |

## ۴- مقایسه نتایج استفاده از BERT برای Text Classification:

همه ی نتایج که در قسمت ۳ اشاره شد را در قالب یک جدول می آوریم تا بتوانیم مقایسه خوبی انجام دهیم.

| ***Model Number*** | ***Accuracy on Test data*** |
|---|---|
| Model 3-1 (ParsBERT) | 0.8831882730187814 |
| Model 3-2 (ParsBERT) | 0.8547869903802107 |
| Model 3-3 (ParsBERT) | 0.8886852954649564 |
| Model 3-4 (ParsBERT) | 0.8992212551534585 |
| Model 3-5 (Multi lingual) | 0.8776912505726066 |

## ۵- جمع بندی و مقایسه – BERT یا tf-idf؟

همانطور که در تمرین شماره ۴ ذکر شد در حالتی که از SVM با کرنل rbf استفاده کردیم بیشترین دقت را از مدل برای داده تست که تقریبا برابر **۸۵ درصد** بود گرفتیم. این دقت با توجه به داده های ما و اینکه با استفاده از SVD تجزیه انجام میدهیم قابل قبول است. اما در مقایسه با شبکه های عصبی از پیش آموزش داده شده BERT همانطور که در بالا میبینید اختلاف درصد ها فاحش است و دقت **۹۰ درصد** گرفته ایم. میتوانیم نتیجه بگیریم در شبکه های BERT معمولا دقت های بیشتری به نسبت روش های آماری مثل tf-idf قابل دریافت است. این در حالی است که دقت هایی که در این تمرین گرفتیم قابل بهبود هستند و با تغییر پارامتر های مدل میتوانند بهبود پیدا کنند.