






فایل‌های ضمیمه :

 BERTmultilingual.ipynb
 DistilBERT.ipynb
 parsBERT.ipynb
 prepare_data.ipynb
 TextAttack.ipynb

قدم اول : داده‌های persica.csv را به نسبت ۸ به ۲ به مجموعه ی تست و آموزش بخش می‌کنم. داده‌هایی که به عنوان متن مورد بررسی قرار می‌گیرد ترکیب هنوان و متن خبر هست که در تمرین قبل هم مورد استفاده قرار گرفت. حالا در میان داده‌های آموزش باز هم نسبت ۲ به ۸ برای داده های اعتبارسنجی در فرایند آموزش و تنظیم وزن‌ها جدا می‌کنم. برای اینکه آموزش هر یک از شبکه ها زمان زیادی لازم دارد و برای اینکه از تکرار مجدد این تقسیم بندی در هر بار اجتناب کنم، داده های تقسیم شده را در سه فایل train.csv, test.csv, validation.csv ذخیره کردم و این فایل ها برای fine-tuning تمام مدل های به کار رفته در این گزارش استفاده شده اند. ستون هایی از فایل اصلی persica.csv که به تمرین غیرمرتبط بود هم در این فایل ها نیامده. علاوه بر این چون خروجی شبکه ها احتمالاتی است که به هریک از برچسب ها اطلاق می‌کند برای راحتی کار برچسب ها را به عدد تبدیل کردم تا هر عدد اندیس احتمال نظیر شده به آن در بردار خروجی شبکه باشد و هم از نظر مفهوم و هم به لحاظ اعتبارسنجی کار راحت تر شود. در فایل prepare_data.ipynb این عملیات انجام شده.

قدم دوم : انتخاب شبکه‌هایی که قرار است استفاده شوند و برای دسته بندی اخبار به ۱۱ کلاس مجزا fine-tune بشوند. من از ۴ مدل BERT استفاده کردم. تمام مدل‌ها مدل های زبانی هستند ینی که شبکه هایی هستند که صرفا با ماسک شدن بخشی از ورودی هایشان و سپس پیش بینی آن بخش ماسم شده به صورت self supervised آموزش دیده اند و هنوز برای هیچ گونه کاربردی fine-tune نشده اند. از سایت <https://huggingface.co> مدل‌ها انتخاب شده‌اند.

- textattack/bert-base-uncased-ag-news : یک شبکه که با داده های اخبار انگلیسی آموزش دیده و مدل‌های fine-tune شده اش طبق گزارش هایی که آمده نتایج خوبی برای دسته بندی متن ارایه کرده است.
- distilbert-base-uncased : یک مدل که با داده‌های ویکیپدیا و کتاب و مجلات انگلیسی آموزش دیده.
- HooshvareLab/bert-fa-zwnj-base : همان مدل برت فارسی یا parsbert هست.
- bert-base-multilingual-cased : یک مدل زبانی که با داده های ۱۰۴ زبان مختلف آموزش دیده.

مراحل بعدی برای هر یک از این مدل ها در یک فایل مجزا انجام شده. این فایل ها روی colab اجرا شده اند و به دلیل محدودیت زمانی برای استفاده از منابع، همه در یک فایل قرار ندارند.

قدم سوم : فایل‌های آماده شده را به صورت کلاس های dataset باز میکنم تا پیش پردازش ها روی آن انجام شوند. متن ورودی با استفاده از شبکه انتخاب شده واحدسازی شود. در این عملیات تعداد ورودی با اندازه ی دهانه ی شبکه تنظیم میشود، به آن حاشیه اضافه میشوند یا از آن خذف می‌شود و هر حمله به کلمه های سازنده اش بخش میشود و خروجی این مرحله شامل

داده هایی ست که به طور کامل پیش پردازش شده اند و آماده ی ورودی به لایه اولیه شبکه هستند. انکودینگ موقعیت ها و attention برای لایه ورودی محاسبه میشود.

قدم چهارم : تنظیم پارامترها برای fine-tuning انجام میشود. برای هریک از شبکه ها دسته های ۸ یا ۱۶ تایی با ایپاک هایی در اندازه های ۳ تا ۱۰ انتخاب شدند. البته معمولاً برای این کار عدد ۳ یا ۴ را در نظر میگیرند. نرخ یادگیری را در هر مورد بین $1e-3$ تا $1e-5$ جابجا کردم. و بارها شبکه ها را فاین تیون کردم تا نتایج بهتری بگیرم. بهترین مدلی که از هر کدام بدست آمد را در فایل ها آوردم. به دلیل حجم بالای مدل ها فقط کد حاوی نتیجه ی اجرای آن ها ضمیمه شده. نتایجی که در ادامه گزارش میشود بر اساس نرخ یادگیری ۰,۰۰۰۰۱ و با ۳ ایپاک حساب شده اند. در مورد parsbert این منیجه راصی کننده بود ولی در مورد باقی مدل ها با تغییراتی که در پارامترها ایجاد کردم تغییر محسوسی در کارایی مدل ایجاد نشد. سه مدل دیگر اصلاً عملکرد درستی روی داده ها نشان ندادند و مقدار loss حتی در گردش های ۱۰ تا ۱۵ هم نشانه ی مشخصی از همگرایی نشان نداد.

قدم پنجم : بعد از اینکه fine-tuning تمام شد مثل تمرین قبلی برای داده های تست مقدار f1, recall, precision را حساب کردم. که در زیر آورده شده.

| | PRECISION | RECALL | F1 SCORE |
|--------------------------------------|-----------|--------|----------|
| textattack/bert-base-uncased-ag-news | 9 | 1 | 1 |
| distilbert-base-uncased | 14 | 4 | 5 |
| HooshvareLab/bert-fa-zwnj-base | 89 | 89 | 89 |
| bert-base-multilingual-cased | 9 | 1 | 1 |

همان طور که قبلاً هم اشاره شد تنها مدل قابل قبول و قابل استفاده اینجا parsbert هست. باقی مدل ها احتمالاً برای مدلسازی ساختار زبان فارسی مشکل داشته اند. یا مثلاً توکن سازی کلمات ورودی را اصلاً نتوانسته اند به خالت مطلوبی انجام دهند. در تمام اجراها خروجی مدل این است که یک برچسب را برای تمام داده های تست خروجی می دهد در واقع اصلاً دسته بندی در کار نیست.

حالا نتایج تمرین قبل را در زیر می اورم :

| | Naive_Bayes | Perceptron | SVM |
|-----------|-------------|------------|------|
| precision | 0.77 | 0.76 | 0.82 |
| recall | 0.78 | 0.78 | 0.82 |
| f-score | 0.77 | 0.76 | 0.82 |

هر سه مدل کلاسیکی که روی ماتریس tf-idf تمرین قبلی که تبدیل lsa روی آن انجام شده بود نتایج قابل قبولی میگیرند ولی مدل SVM نتیجه ی بهتری گرفته. این مدل را با parsbert قیاس کنیم. استفاده از یک شبکه ی برت مناسب معیارهای اعتبارسنجی را به طور قابل توجهی بهتر کرده.