به نام خدا

گزارش تمرین سوم

گروه خیاط

سعید هدایتیان، علی مرعشیان، مریم قیصری

تعریف مسئله : کامل کردن کلمه ی جاری در جمله برای زبان فارسی (auto complete Persian word)

یکی از کار برد های این مسئله وقتی است که کلمه ای را روی تلفن خود می نویسید و مشاهده می کنید کلماتی که می توانند با آن کلمه مطابقت داشته باشند بلافاصله ظاهر می شوند. در واقع برنامه ای وجود دارد که سعی می کند تا فهرستی از کلماتی با احتمال بیشتر ارائه کند که به بهترین وجه با اجزای جمله ارتباط داشته باشد. برای تولید کلمات براساس احتمال، موضوع خاصی در NLP به نام مدل‌های زبانی مطرح می شود.

یک مدل زبانی در واقع توزیع احتمال بر روی توالی کلمات است که با توجه به دنباله ای از کلمات به طول m، یک احتمال P(W1….Wm) را به کل دنباله اختصاص می دهد.

در این جا ما ابتدا برای پیشبینی کلمات جایگزین کلمه ی ناقص (mask) از یک مدل پیش آموزش روی متون زبان فارسی به نام ALBERT که مبتنی بر Bert transformer می باشد استفاده می کنیم و برای انتخاب بهتر کلمه ی محتمل، از مدل n-gram کمک می گیریم که جلوتر توضیح مختصری از عملکرد هر یک خواهیم داد.

دو فایل transformermodel.py و ngrammodel.py به ترتیب مدل ALBERT (A lite BERT for self-supervised learning of language representation) و N-gram می باشند.

- **ALBERT**

طرز کار مدل‌های مبتنی بر autoencoder (BERT) به این صورت است که آن ها با خراب کردن token های ورودی و تلاش برای بازسازی جمله اصلی از قبل آموزش داده می‌شوند. آنها با encoder مدل اصلی transformer مطابقت دارند به این معنا که بدون mask به ورودی های کامل دسترسی دارند. این مدل‌ها معمولاً یک representation دو طرفه از کل جمله می‌سازند. آنها را می توان به خوبی fine-tune کرد و در بسیاری از کارها مانند کامل کردن کلمات، به نتایج عالی دست یافت.

- **N-gram**

همچنین برای شرح مختصری از نحوه ی کار مدل های N-gram ای چنین اند که برای هر کلمه ی تعیین شده در جمله احتمال زنجیره ای (Markov chain) آن را بر اساس N کلمه ی قبلی به دست می آورد و تنها این مقدار به کلمات قبلی وابسته بوده و کلمات بعد از آن را در نظر نمی گیرد. و حال بر اساس این مقادیر می توان تصمیم گرفت که کدام کلمه محتمل تر است. در مدل ما N از ۱ تا طول جمله حرکت کرده و سپس بر اساس مقادیر احتمال کلمات را مرتب کرده و کلمه ای با score بیشتر را برمی گرداند.

در واقع در مدل ما K کلمه ی ابتدایی توسط مدل اولیه که مبتنی بر BERT می باشد پیشبینی شده بر اساس score مرتب می شوند و سپس احتمال به دست آمده از مدل N-gram تصمیم میگیرد که کدام یک از کلمات محتمل تر است.

- **Data preprocessing and normalizing**

داده هایی که برای این تمرین انتخاب کردیم مربوط به متون خبری موجود در دیتاست cultural می باشد که ما تنها ۱۰ درصد آن را پردازش می کنیم و از این مقدار۱۰درصد برای داده ی تست و ۹۰ درصد مربوط به داده ی آموزش می باشد.

قبل از آن که داده را به مدل بدهیم نیاز است تا آن ها را پیش پردازش نماییم.

از این رو در کلاس Autocomplete موجود در فایل autocomplete.py در ابتدا کد هایی جهت آماده سازی داده ها وجود دارد. که به ترتیب زیر عمل می کند.

۱- جایگزین کردن کارکتر های استاندارد با کارکتر های غیر استاندارد نظیر: "ۀ"، "ؤ"، "إ"، "ئ"، "ﻷ" و ..
۲- جایگزین کردن تمامی space ها و hyphens ها با white space ها
۳- حذف برخی از علائم و کارکترهای نگارشی در این نوع مسئله اهمیتی ندارند و برای سادگی بهتر است نادیده گرفته شوند، مانند : ‘\’ , 'η' , 'ζ', 'ε', 'β', 'α',
۴- و همچنین merge کردن جملات موجود در یک پارگراف.

و سپس با استفاده از تابع Normalizer از کتابخانه ی HAZM متن پردازش شده را نرمالایز می کنیم و برای تسک اصلی آماده شده است.

برای آن که distance پیشوند کلمه ای که کامل نیست و هدف کامل کردن آن است را با پیشوند کلمات پیشنهادی به دست بیاوریم تا بتوان تعدادی از آن ها که تشابهی با پیشوند ندارند را حذف کرد، در این فایل تابعی به نام prefix_distance وظیفه ای محاسبه ی distance را دارد. برای مثال کلمه ای که باید پیش بینی شود دارای پیشوند "اسل" می باشد و اصل کلمه "اسلامی" بوده و همچنین کلمه ی پیشنهادی "سلیم" که پیشوند

آن "سلی" می باشد است، این تابع این فاصله را دو به دو برای کلمه ی اصلی و کلمه ی پیشنهادی اندازه گرفته و K تای با کمترین فاصله انتخاب می شوند.

- **Models' config**

برای مدل های از پیش آموزش دیده فایل هایی نظیر دو فایل tokenizer.json و tokenizer_config.json فولدر models_dir وجود دارند که اطلاعاتی حاوی انواع token ها و مقادیر عددی که جای گزین token ها متن آموزش دیده می شوند، وجود دارد.

و فایل هایی که در هنگام fine-tune و سپس آموزش مدل شامل تعداد زیادی دیکشنری که باید ذخیره شود و مجدد برای تست آن فایل مربوطه load شوند. فایل pytorch_model.bin حاوی این مقادیر می باشد.

همچنین مقادیر برخی از پارامتر های مورد نیاز مدل مانند مقدار model , model type , learning rate version و ....در فایل config.json موجود است.

- **Transformer models**

کلاس BertAutoComplete که از کلاس Autocomplete مشتق شده است به این صورت عمل کرده که مقادیری را به attribute های خود نسبت داده و بر اساس آن ها مقدار های پیش فرض و مورد نیاز برای مدل را تنظیم می کند. پس از آن با تعریف متود complete تعیین می کند که تعدادی از کلمات پیشنهادی را به همراه مقادیرآن ها با کمک متود top_k به دست آورده و edit distance آن ها که متود آن در قسمت پیش پردازش تعریف شده بود را محاسبه کرده و سپس ۵ تا از بهترین کلمات را پیشنهاد می کند.

پس از آن دیتاست مورد نیاز با استفاده از متود create_dataset، به همان نسبت test و train از پیش تعیین شده جداسازی شده، طول جملات آن ها را یکسان کرده، batch هایی به طول ۱۰۰۰ ایجاد شده و دیتاست را می سازد.

حال در متود train مدل از پیش آموزش دیده fine-tune شده و آموزش می بینید و معیار هایی که برای ارزیابی مدل تعیین شده نظیر perplexity قبل و بعد از آموزش در هر epoch نمایش داده می شود و پس از اتمام آموزش مدل آموزش دیده ذخیره می شود.

برای ارزیابی مدل در متود evaluate به سراغ داده های validation رفته و سپس با حذف کلمه ی قبل از کلمه ی ناقص از جمله لیست کلمات جمله را به تابع complete داده و متغیری تعریف می کنیم که تعداد دفعات پیدا شدن کلمه ی ground truth (یکی قبل از کلمه ی ناقص) در کلمات برگشت داده شده

از متود complete را برطول جمله تقسیم کرده و باز گرداند تا به کمک آن بتوان بهتر عملکرد مدل را ارزیابی نمود.

- **Ngram model**

مشابه قبل در این قسمت نیز ابتدا آرگومان های مربوطه ست می شوند و سپس داده ها به جملات تبدیل شده و token های مربوط به شروع و پایان جمله به آن ها اضافه شده و پس از tokenization، unigram ها ساخته شده و بر اساس آن ها سایر ngram ها ساخته می شود.

برای پیشبینی کلمات بر اساس مقادیر احتمالاتی به دست آمده از ngram ها به دو صورت عمل می کنیم:

۱- Back off:

به این صورت است وقتی میخواهد کلمه ای را پیشبینی کند ابتدای n کلمه ی آخر جمله را در نظر می گیرد و اگر ngram از قبل در دیکشنری های load شده از train مدل روی داده ی train بوده، وجود داشت همان ngram را با بیشترین احتمال به عنوان خروجی بر میگرداند و اگر ngram در دیکشنری ها موجود نبود n-1gram تا 1gram را به همین ترتیب بررسی می کند تا بلاخره بتواند آن کلمه را در دیکشنری ها پیدا کند.

۲- Interpolate:

در این بخش تمامی ngram ها تا unigram ها بررسی می شوند و به هر یک از آن هایی که در دیکشنری وجود دارند مقدار وزنی نسبت داده می شود و در آخر کلمه ی با بیشترین وزن پیشنهاد می شود.

باقی متود های موجود در این فایل مشابه متود های بخش قبلی می باشد.

- **Evaluated model based on some sample**

# Epoch	Training time	Perplexity before training	Perplexity after training
3	1.5 h	40.44	30.40
10	4.5 h	39.12	28.49
5	2.5 h	41.3	29.34

دقت مدل Bert روی داده های test حدودا 58 درصد و برای مدل ngram حدودا 33 درصد می باشد.
و همچنین با بررسی loss در هر epoch برای داده های train و validation از صحت عملکرد مدل مطمئن شده که مدل overfit نکرده است.
حال جملاتی را که برای تست به مدل BERT دادیم و خروجی هایی را که از آن گرفته ایم را در زیر می بینید:

```
...برای تهمه ی این ها دلایلی وج
['وجود', 'وجه', 'وجدان', 'وجود', 'وج']
...ون توجه به ادامه ی این قسمت می
['می', 'میباشد', 'میپردازیم', 'میشود', 'میگه']
...رای فرهنگسازی در بین مردم این اقدامات را گست
['گستر', 'گستر', 'گسترش', 'گستردگی', 'گستاخ']
همه ی معاونین رئیس جم... در جلسه حضور داشتن
['جمهور', 'جمهوری', 'جمهوری', 'جمعیت', 'جمله']
.ردم با رعایت قوا... می توانند نقش مهمی در بهتر شدن جام... داشته باشند
['قوانین', 'قواعد', 'قوانینی', 'قوانین', 'قوا']
ال با ازدیاد جمعیت مشک... بیشتری وجود خواهد آمد
['مشکل', 'مشکلات', 'مشکلاتی', 'مشکلی', 'مشکوک']
```

خروجی مدل ngram به ازای هر یک از جملات بالا:
مشاهده می کنید که این مدل کلمات نزدیک تر را پیشبینی کرده و در همه ی این مثال ها توانسته در لیست کلمات پیشنهادی کلمه ی اصلی را هم درست حدس بزند.

```
...برای تهمه ی این ها دلایلی وج
['وجود', 'و', 'وارد', 'وی', 'اجرا']
...بدون توجه به ادامه ی این قسمت می
['میان', 'میزان', 'فیلم', 'مراسم', 'موضوع']
...برای فرهنگسازی در بین مردم این اقدامات را گست
['است', 'استقبال', 'دسترسی', 'هستند', 'اساسی']
.مه ی معاونین رئیس جم... در جلسه حضور داشتند
['اما', 'ضمن', 'هم', 'امسال', 'کمتر']
مردم با رعایت قوا... می توانند نقش مهمی در بهتر شدن جام... داشته باشن
['قرار', 'توانسته_است', 'و', 'یا', 'اما']
با ازدیاد جمعیت مشک... بیشتری وجود خواهد آمد
['آید', 'آنکه\u200cمشکلات', 'مشکلاتی', 'مرکز', 'می']
```