دانشگاه تهران

درس پردازش زبان طبیعی
تمرین سری ٦

نام

حسین سیفی

شماره دانشجویی

٨١٠١٠٠٣٨٦

تاریخ ارسال گزارش

١۴٠٠/٠٣/٣١

فهرست مطالب

Q1.Question Answering (QA) Project

هدف این سوال Fine-Tune کردن مدل‌های از پیش آموزش دیده(Pretrained) ParsBERT و AlBERT به کمک مجموعه داده‌های متفاوت از جمله PersianQA ، Pquad و ParSQuAD است به طوری که مدل‌ها پس از پایان Fine-Tunning با دقت مناسبی قادر به پاسخگویی به سوالات(Question Answering) باشند. وظایف مدل‌های پاسخگویی به سوال به این صورت تعریف می‌شود که مدل مد نظر یک متن و سوالاتی را به عنوان ورودی دریافت می‌کند و سپس باید اندیس شروع و پایان بهترین جواب ممکن برای هر سوال در متن ارائه شده مشخص کند. پیش از آن‌که بتوانیم مدل‌های خود را Fine-Tune کنیم، ابتدا نیاز است تا پیش‌پردازش مناسب را بر روی داده‌ها انجام دهیم تا متن، سوالات و پاسخ‌ها با فرمت مناسب به مدل‌ها داده شوند.

پیش پردازش

فرمت فعلی داده‌ها برای Fine-Tune کردن مدل‌ها مناسب نیست و باید با کمک پیش پردازش مناسب تمامی داده‌های موجود را به یک فرمت مشخص تبدیل کنیم. ابتدا دو مجموعه داده PQuAD و ParSQuAD یک مرحله اضافی را نسبت به PersianQA می‌پیمایند تا از قالب Json به یک دیتافریم از کتابخانه Pandas و سپس در قالب یک Object از کتابخانه DataSet قرار بگیرند.

پس از انجام مراحل فوق و قرار گرفتن هر یک از سه مجموعه داده مورد نظر در شرایط یکسان، داده‌های ستون‌های Context و Question که به ترتیب شامل متون و سوالات هستند را به یکدیگر الصاق می‌کنیم و سپس باید متن و سوالات به هم چسبیده را توکنایز کنیم. توکنایز کردن این متون به کمک توکنایزر بارگیری شده هر یک از مدل‌های از پیش آموزش دیده انجام می‌شود. در ادامه هر یک از پارامترهای استفاده شده در توکنایزر معرفی می‌شوند:

۱. پارامتر Max_length: این پارامتر حداکثر اندازه بردارها را مشخص می‌کند. مقدار انتخاب شده با مراجعه به مقالات و سایت‌های مختلف و بررسی بهترین مقادیر ممکن، عدد ۳۸۴ می‌باشد.
۲. پارامتر Truncation: این پارامتر مشخص می‌کند که آیا بردارهای ایجاد شده که طولی بیشتر از حداکثر اندازه ممکن دارند تقطیع شوند و در چند بردار قرار بگیرند یا خیر؟ مقدار این پارامتر برابر Only_second انتخاب شده است و بدین معناست که فقط بخش دوم ورودی که همان Context یا متن است اجازه تقطیع دارد.
۳. پارامتر Padding: همانطور که می‌دانیم بردارهای ورودی مدل‌های مبتنی بر برت باید طول یکسانی داشته باشند و به وسیله مقدار max_length برای پارامتر padding طول بردارهای کوتاه‌تر از حداکثر طول را به اندازه مد نظر می‌رسانیم.

خروجی توکنایزر با پارامترهای فوق سه بردار برای هر یک از ورودی‌ها خواهد بود که این سه بردار با نام‌های input_ids، attention_mask و token_type_ids در داده‌های پردازش شده قابل مشاهده هستند.

بخش Answers در مجموعه داده شامل دو بخش answer_start و text است که اولی اندیس شروع پاسخ مناسب را در متن ارجاع شده نشان می‌دهد و بخش دوم متن‌های ممکن برای پاسخ سوال داده شده را نشان می‌دهد اما این فرمت به عنوان ورودی شبکه عصبی مبتنی بر برت قابل قبول نیست. دو ستون که جایگزین answers می‌شود، ستون‌های start_position و end_position هستند که شروع و پایان جواب‌های مناسب را نشان می‌دهد.

تمامی عملیات فوق به کمک تابع preprocess_function انجام می‌شود و با اعمال این تابع بر روی داده‌های آموزش و اعتبارسنجی، داده‌ها آماده ورود به شبکه عصبی خواهند بود.

ایجاد مدل

سپس مدلی از نوع مورد نیاز(در بخش اول ParsBERT و در بخش دوم Albert) بارگیری می‌شود و با بهترین مقداردهی ممکن به پارامترها، مدل را ایجاد و Train می‌کنیم. پارامترهای استفاده شده به شرح زیر می‌باشند:

۱. Learning_rate: این پارامتر نرخ یادگیری را برای مدل تعریف می‌کند که مقدار آن همانند بسیاری از مدل‌های پردازش زبان طبیعی برابر $3 * 10^{-5}$ در نظر گرفته شده است.
۲. Evaluation_strategy: از بین دو استراتژی که بیشتر برای ارزیابی استفاده می‌شوند(Case و Epoch)، از نوع Epoch برای این تمرین استفاده می‌شود.
۳. num_train_epochs: تعداد Epochهای انتخاب شده برای این تمرین برابر ۳ می‌باشد چرا که با بررسی‌های انجام شده مشخص شد اکثریت مدل‌ها پس از ۲ Epoch به مشکل Overfitting برمیخورند و انتخاب مقداری بین ۲ و ۳ برای این پارامتر می‌تواند ایده‌آل باشد.
۴. Batch_size: مقدار این پارامتر برای بخش ارزیابی و آموزش به طور مساوی برابر ۱۶ در نظر گرفته شده است.

پس از Fine-Tune شدن مدل، می‌توان اطلاعات خروجی را در قالب سه بردار Loss، Start_logits و End_logits برای هر یک از ورودی‌های داده شده دریافت کرد که کاربرد هر یک با توجه به نام آن مشخص است.

ارزیابی

تنها داده‌ی تست موجود در مجموعه متعلق به مجموعه PQuAD است که برای ارزیابی سایر مدل‌ها نیز استفاده می‌شود. ابتدا پیش پردازشی مشابه آنچه بر روی داده‌های آموزش و اعتبارسنجی اعمال شد را بر روی داده‌های تست نیز پیاده سازی می‌کنیم و با استفاده از تابع Predict پاسخ مدل را برای داده‌های تست دریافت می‌کنیم. در نهایت با پس پردازش جزئی و یکسان سازی فرمت داده‌های خروجی تابع predict و داده‌های طلایی، می‌توان معیارهای ارزیابی را با تابع Compute از متریک squad نسخه ۲ که بارگیری شده است، محاسبه کرد. مقادیر معیارهای ارزیابی exact match و f1 score و validation loss برای هر یک از مدل‌های Fine-tune شده با داده‌های متفاوت به شرح زیر می‌باشد:

PARSBERT

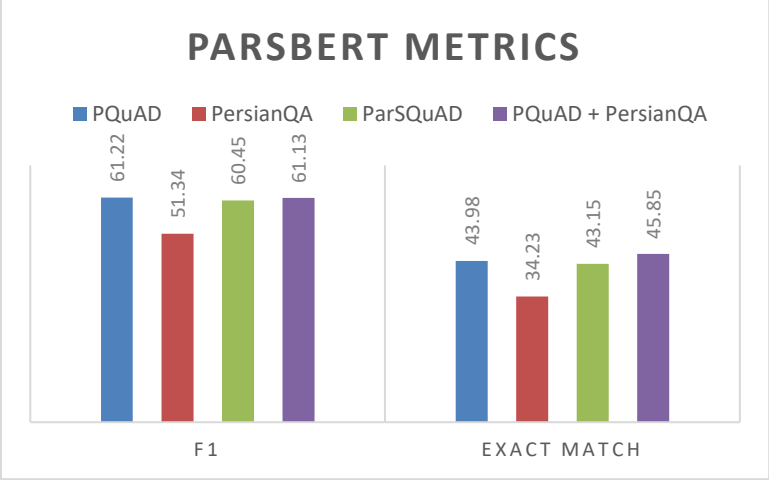

Figure 1

1.PQuAD QA model (use PQuAD dataset)

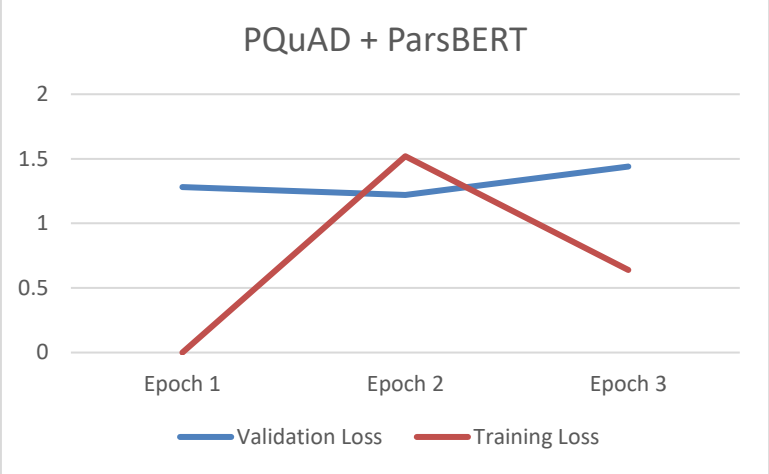

Figure 2

2. PersianQA QA model (use PersianQA dataset)

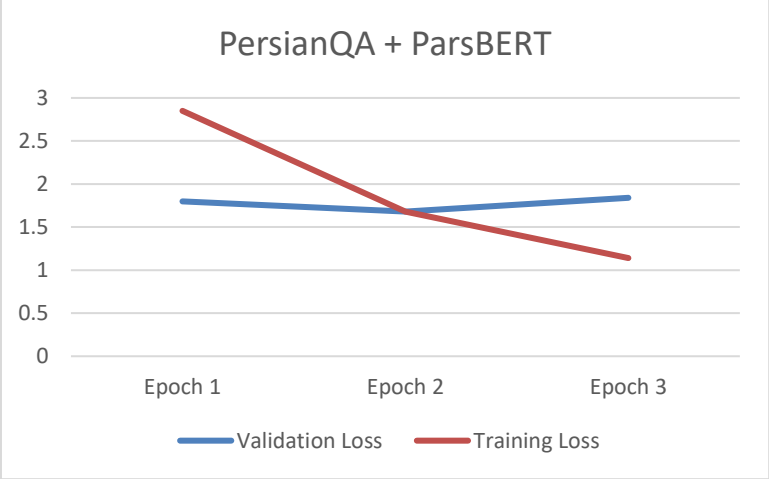

Figure 3

3. ParSQuAD QA model (use ParSQuAD dataset)

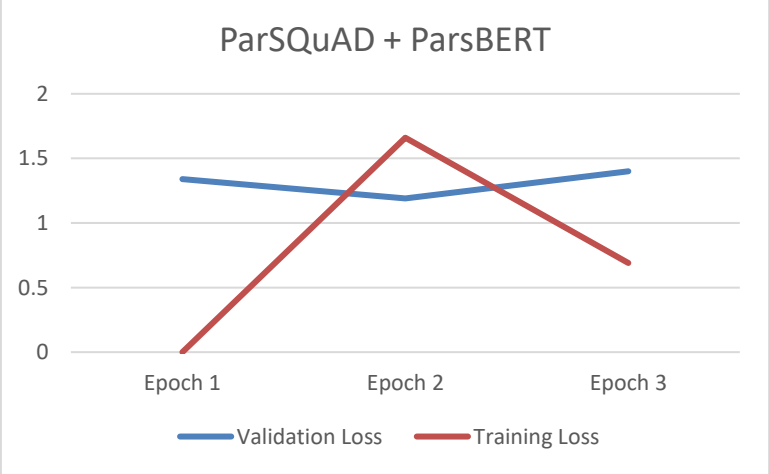

Figure 4

4. PQuAD and PersianQA QA model (use mix of PQuAD and PersianQA datasets)

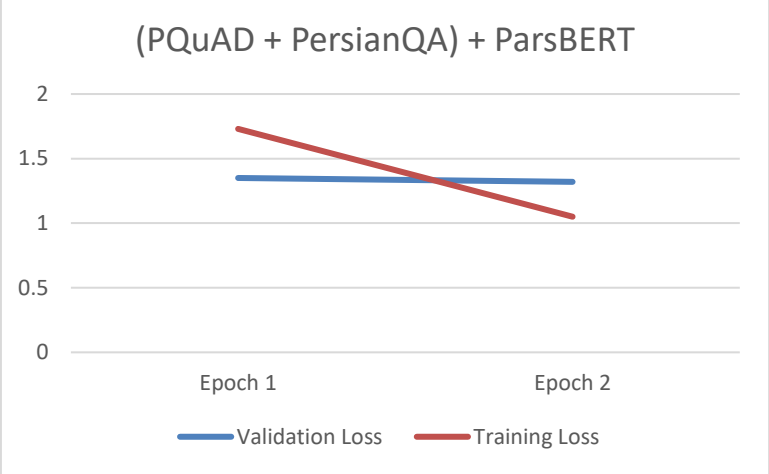

Figure 5

ALBERT

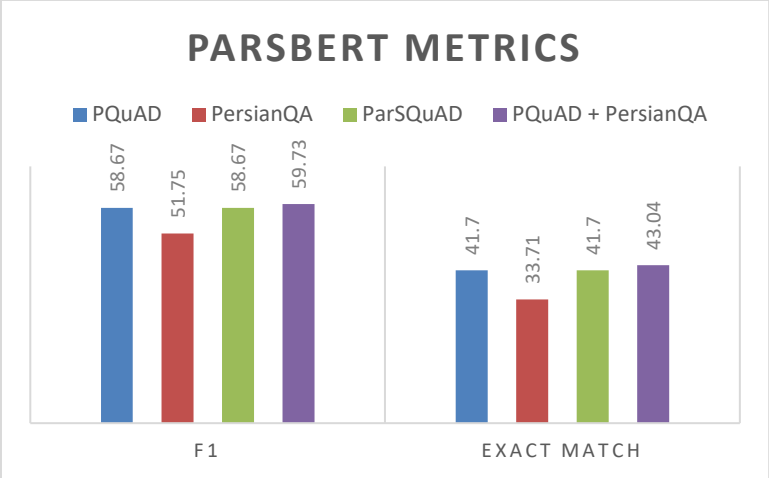

Figure 6

1.PQuAD QA model (use PQuAD dataset)

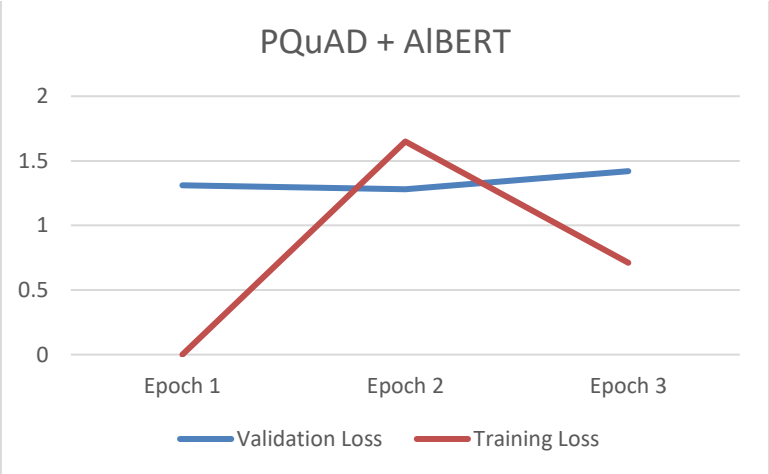

Figure 7

2. PersianQA QA model (use PersianQA dataset)

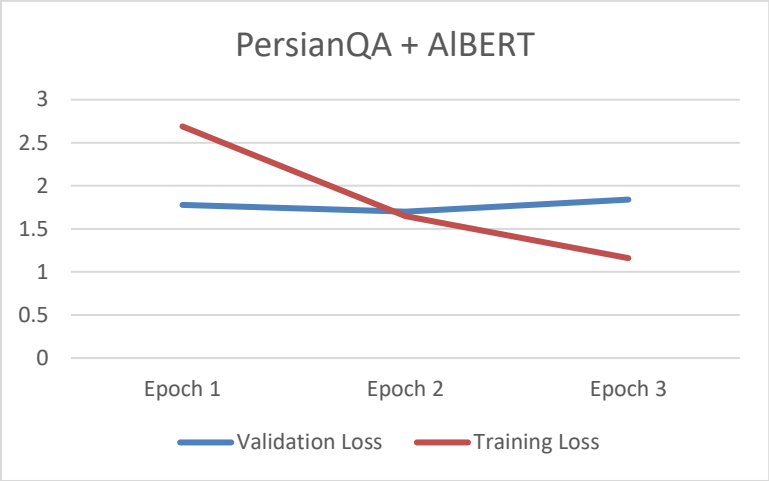

Figure 8

3. ParSQuAD QA model (use ParSQuAD dataset)

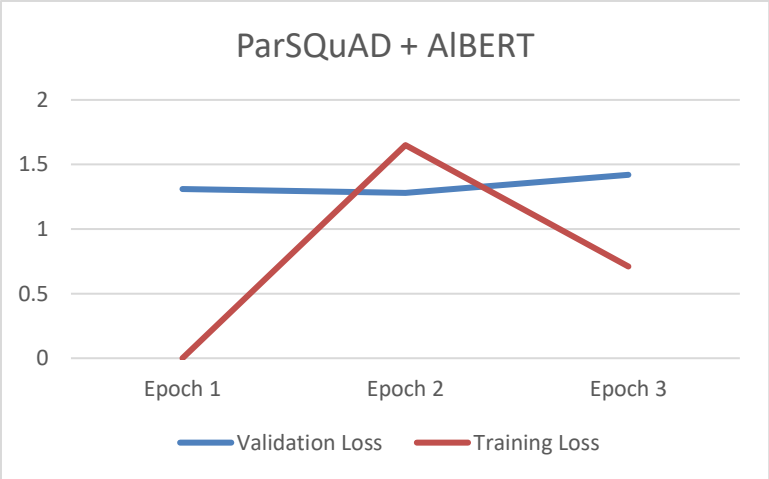

Figure 9

4. PQuAD and PersianQA QA model (use mix of PQuAD and PersianQA datasets)

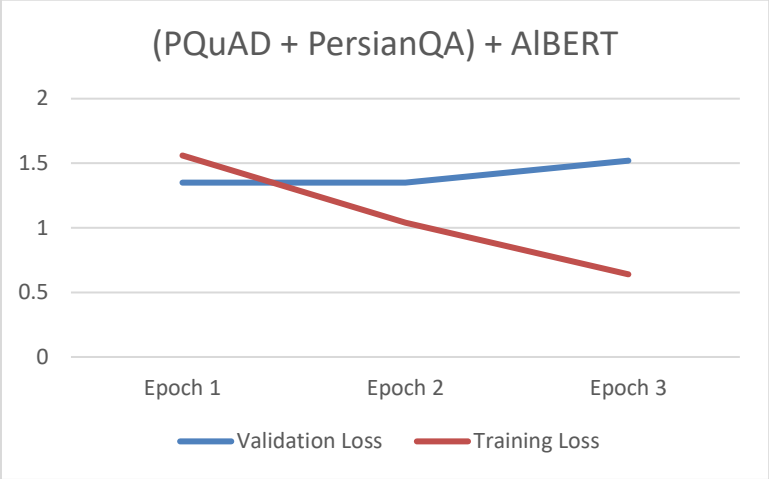

Figure 10

نتیجه

همانطور که در نمودارهای فوق قابل مشاهده است با کاهش مداوم مقدار Training Loss و افزایش مقدار Validation loss پس از Epoch شماره ۲، مشخص می‌شود که تمامی مدل‌های Fine-Tune شده پس از Epoch دوم دچار Overfitting شده‌اند و به کمک متدهای Early stopping امکان توقف زودهنگام آن‌ها وجود دارد اگر چه این عمل تنها در یکی از مدل‌های Fine-Tune شده انجام گرفت.

به طور کلی با بررسی ۸ مدل موجود قابل توجه است که مدل‌هایی که از مدل Pretrained اول یعنی ParsBERT استفاده کرده‌اند عملکرد بهتری در معیارهای Exact match و F1 نسبت به AlBERT داشته‌اند و همچنین ترکیب دیتاست‌های PersianQA و PQuAD برای Fine-tune کردن مدل ParsBERT بهترین نتیجه ممکن را ارئه می‌دهد. از طرف دیگر مقادیر معیارهای ارزیابی برای مدل‌هایی که از PersianQA به تنهایی استفاده کرده‌اند ضعف چشم‌گیری نسبت به دیگر دیتاست‌ها دارند که می‌تواند به دلیل عدم تطابق مجموعه تست با مجموعه استفاده شده به عنوان مجموعه آموزش باشد اما با بررسی مقادیر Loss در این مدل و مشاهده اختلاف بسیار زیاد Loss مدل Fine-tune شده به کمک این مجموعه با دیگر مجموعه‌ها می‌توان به این نتیجه رسید که مجموعه داده PersianQA نسبت به دیگر مجموعه‌های داده‌ای استفاده شده برای Question Answering مجموعه‌ای ضعیف‌تر است و دیتاست‌هایی مانند PQuAD و ParSQuAD نسبت به این دیتاست ارجحیت دارند.

در نهایت با توجه به پیچیدگی بالای مسائل Question Answering و حتی دقت پایین انسان در پیاده سازی این وظیفه و کوچک بودن دیتاست‌های در دسترس، ارزیابی‌های انجام شده بر روی مدل‌های فوق عملکردی نسبتا مطلوب را در سطح خود به نمایش می‌گذارند.

Q2. Natural Language Understanding Projects

abstract and introduction

هدف:پردازش زبان طبیعی بعد از Fine-tune کردن یک مدل شبکه عصبی که از قبل pretraine شده است، برای اینکه گفتاری که کاربر در زبان فارسی میدهد را به representationی تبدیل کند که به اندازه کافی ساختاریافته باشد که توسط یک سرویس خودکار پردازش شود.

اینتنت به نیتی که متن دارد، اشاره میکند. Intent classification یک مساله کلاسبندی ساده در سطح جمله است که یک sentence level class label بهمون میدهد . ساخت یک کلاسیفایر در سطح جمله از طریق fine-tune کردن یک مدل Transformer-base که pre-traine شده انجام میشود.

روش هایی که باهاشون میشود یک مساله sequence labeling را حل کرد Maximum Entropy Markov Models (MEMMs), Conditional Random Fields (CRFs), and Recurrent Neural Networks (RNNs) هستند.

اسلات به عناصر کلیدی در جستجوی کاربر اشاره میکند.Slot Filling یک مساله کلاسبند در سطح توکن یا کلمه است که از BIO-annotation استفاده میکند که از تگ هایBIO به عنوان کلاسهای تارگت استفاده میکند(انتظار داریم یک دنباله ای از named entity ها در جمله بعنوان کلاسها بهمون برگردونه). Slot های به درد بخور(typed named entities) را از جمله استخراج میکند. این بخش مثل Named Entity Recognition در NLP literature عمل میکند.

خلاصه Intent classification و Slot Filling به عناصرکلیدی در جمله ورودی اشاره دارند. هدف ما ساختن مدلی ست که هر دوی این مسائل را داشته باشد. BERT راه حل خوبی ست که سیستم SLU / NLP را ساده سازی میکند که فقط با همین یک مدل میتوان عمل trained و fine-tuned را انجام داد.

Data

data با استفاده از خود ابزار فایل ساخته میشود. بعد تبدیل به یک فایل دیتاست هاگینگ فیس میشود، بعد دیتای فارسی در content لود شده و ازش استفاده میشود. فقط دیتای فارسی برداشته میشود زیرا دیتاست داده شده در قالب فایل های جیسون است که در ۵۱ زبان مختلف سازماندهی شده اند. یک خط از فایل جیسون در زبان فارسی به شکل زیر است و آرگومانهای آن نیز تشریح شده اند:

{
"id": "0",
"locale": "fa-IR",
"partition": "test",
"scenario": "alarm",
"intent": "alarm_set",
"utt ":" این هفته ساعت پنج صبح بیدارم کن" ,
"annot_utt" : date ساعت ["این هفته"] : time :["پنج صبح بیدارم کن"],
"worker_id": "8",
"slot_method": [
{
"slot": "time",

```
"method": "translation"
},
{
"slot": "date",
"method": "translation"
}
],
"judgments": [
{
"worker_id": "3",
"intent_score": 1,
"slots_score": 1,
"grammar_score": 4,
"spelling_score": 2,
"language_identification": "target"
},

 {
"worker_id": "21",
"intent_score": 1,
"slots_score": 1,
"grammar_score": 4,
"spelling_score": 2,
"language_identification": "target"
  },
 {
"worker_id": "8",
"intent_score": 1,
"slots_score": 1,
"grammar_score": 4,
"spelling_score": 2,
"language_identification": "target"
  }
 ]
}
```


۱. id همان شناسه اصلی در مجموعه SLURP است.
۲. Locale کد کشور و زبان گفتگوها
۳. partition همان train, dev یاtest طبق تقسیم بندی اورجینال در SLURP است.
٤. scenario دامنه general یک utterance است که در SLURP با نام "سناریو" شناخته می شود
٥. intent محتوای کل متن را خلاصه میکند
٦. utt متن گفتگو را نشان میدهد البته بدون حاشیه نویسی.
۷. annot_utt متن گفتگو با slot annotation .
۸. worker_id هر منطقه یک worker id خاص خودش را دارد که برایلوکالیزیشن ازش استفاده میشود.

۹. slot_method برای هر اسلات از گفتگو یک متد وجود دارد. این اسلات متد میتواند translation، localization یا unchanged باشد. اگر unchanged باشد اسلات اولیه بدون هیچ تغییری کپی و استفاده میشود، اگر translation باشد اسلات اولیه همان است فقط به یک زبان دیگر است. اگر localization باشد عبارت متفاوتی انتخاب میشود که برای عبارت موجود در آن جا مناسب تر است.

۱۰. Judgments هر Judgment برای محل یابی گفتگو دارای ۶ کلید است. worker_id و "intent_score", "slots_score", "grammar_score","spelling_score", "language_identification" که از اسم هایشان مشخص است. مثلا intent_score مشخص میکند چقدر جمله با intent مچ میشوند، slots_score مشخص میکند چقدر همه این عبارت‌ها با دسته‌های داخل پرانتز مطابقت دارند، grammar_score مشخص میکند چقدر جمله طبیعی به نظر میرسد، spelling_score مشخص میکند چقدر املای کلمات درست است البته تفاوت های املایی که ممکن است به دلیل تفاوت در گویش باشد را نادیده میگیرد. language_identification زبان را تشخیص میدهد.

Preparing the Data in datasets format (Apache Arrow)

آماده سازی دیتا در کلاس DatasetCreator انجام میشود. این کلاس برای ایجاد چهار تقسیم در فرمت Huggingface Datasets Apache Arrow دیتاست MASSIVE است. هر دیتاست تقسیم بندی شده دارای ستون های زیر است:

id","locale","utt","annot_utt","domain","intent_str","intent_num","slots"
"_str","slots_num

توابع زیر در این کلاس وجود دارند:

create_datasets(data_path) : تقسیم داده ها با استفاده از data_path در massive set

add_numeric_labels() :ورژن های عدد صحیح از intent ها و slot ها برای مدل سازی را ایجاد میکند

investigate_datasets() : نمونه هفتم را از هر مجموعه داده را برای بررسی چاپ می کند

save_label_dicts(prefix) : نگاشت ها را در نسخه های عدد صحیح لیبل ها ذخیره می کند

save_datasets(out_prefix) : ذخیره دیتاست در out_prefix

قبل از آموزش مدل یک سری پیش پردازش روی دیتاست انجام شده است و ستونهای اضافه حذف شده است. attention mask یک تنسور باینری است که پوزیشن اندیس های پد شده را نشان میدهد برای اینکه مدل به آنها توجه نداشته باشد.

method

Fine-Tune

fine-tune کردن مدل کلاسبندی ParseBERTروی دیتاست با "bert-base-parsbert-uncased" که از قبل pretraine شده است انجام شده است.

Training an Encoder Model

این جا دو بخش داریم که یک مساله کلاسیفیکیشن ساده ست که جمله را بعنوان ورودی میگیرد اینکه اینتنت چیست را برمیگرداند(60تا کلاس اینتنت هست) به همراه یک مساله seq2seq مانند pos که به ازای هر کلمه تگ موبوطه را برمیگرداند. با این تفاوت که به جای تگ های pos از تگ های این مساله استفاده میشود مثلا "فروشگاه رفاه" میشود تگ مکان. طول اسلاتها نیز ۵۷ است.

طرح کلی مساله با استفاده از مقاله ی **https://arxiv.org/pdf/1902.10909.pdf** انجام شده است فقطاینجا مقداری ساده سازی انجام شده است مثلا از crf و تگهایiob استفاده نشده است ولی در انتها به دقت قابل قبولی رسیده ایم.

قبل از آموزش مدل یک سری پیش پردازش روی دیتاست انجام شد است و ستونهای اضافه حذف شده است. attention mask یک تنسور باینری است که پوزیشن اندیس های پد شده را نشان میدهد برای اینکه مدل به آنها توجه نداشته باشد. برای BertTokenizer اگر مقدارش ۱ باشد یعنی باید به آن توجه شود و صفر باشد به معنی این است که این یک مقدار پد است و نباید به آن توجه شود.

(BERT can be easily extended to a joint intent classification and slot filling model)

Intent classification یک مساله کلاسبندی ساده در سطح جمله است و Slot Filling یک مساله کلاسبند در سطح توکن یا کلمه است. به ازای هر جمله یک intent داریم و به ازای هر کلمه هم برای slot filling یک تگ داریم. این دو بخش همزمان با هم توسط مدل BERT انجام میشوند. به همین دلیل شبکه عصبی دوتا خروجی میدهد ، یکی برای intent detection و یکی برای slot filingینی دوتا لایه خطی جدا از هم بالای BERT وجود دارد.

طبق چیزی که خود برت گفته است اندازه ورودی به برت باید ۵۱۲ باشد و اگر از این کمتر باشد نیاز به استفاده از استراتژی padding دارد پس از pad استفاده میکنیم تا جای خالی ۵۱۲ توکن پر شود. خروجی نیز به همین صورت است. مثلا اگر کلمه اول اتاق باشد تگ مکان ، کلمه دوم آبی میشه تگ مثلا صفت ، تا همینجور کلمه اخر جمله. وقتی جمله تمام شود میرسیم به pad ها ، این پد ها نه باید در لاس حساب بشن و نه در دقت ، در محسابه لاس یک ارگومان ignore وجود دارد که جهت نادیده گرفتن خروجی پد هایی که ما اسم تگ شون هم دوباره پد گذاشتیم هستند.

برای محاسبه دقت هم یک تابع get_non_pad تعریف کردیم که اول تگ های پد را جدا میکند تا فقط جملات خام بمونن بعد دقت را حساب میکند.

تابع get_tp تعداد کلاسهایی که (تگ هایی که) مدل درست پیش بینی کرده است را میشمارد تا بعدا برای محاسبه دقت و.. از آن استفاده کند. البته تگ های pad را اینجا هم حذف میکنیم.

(جهت اجرای کد ابتدا دیتای جیسون فارسی را در پوشه fa در content قرار دهید و سپس کد را ران کنید.)

Result

ارزیابی و پیش بینی خروجی ها به شکل زیر است. دقت intent detection طی ۱۰ ایپاک روی GPU با بهینه ساز AdamW و loss fuction، CrossEntropyLoss در هر ایپاک به صورت زیر قابل مشاهده است:

```
100% 720/720 [17:46<00:00, 1.34s/it]
Epoch: 0,training loss: 1.41 , validation loss: 2.94 , validation intent detection accuracy: 0.06
100% 720/720 [18:03<00:00, 1.34s/it]
Epoch: 1,training loss: 0.75 , validation loss: 2.82 , validation intent detection accuracy: 0.12
100% 720/720 [18:01<00:00, 1.34s/it]
Epoch: 2,training loss: 0.49 , validation loss: 2.45 , validation intent detection accuracy: 0.15
100% 720/720 [18:01<00:00, 1.34s/it]
Epoch: 3,training loss: 0.35 , validation loss: 1.94 , validation intent detection accuracy: 0.26
100% 720/720 [18:01<00:00, 1.34s/it]
Epoch: 4,training loss: 0.27 , validation loss: 1.88 , validation intent detection accuracy: 0.29
100% 720/720 [18:01<00:00, 1.34s/it]
Epoch: 5,training loss: 0.21 , validation loss: 1.58 , validation intent detection accuracy: 0.39
100% 720/720 [18:01<00:00, 1.34s/it]
Epoch: 6,training loss: 0.17 , validation loss: 1.45 , validation intent detection accuracy: 0.45
100% 720/720 [18:01<00:00, 1.34s/it]
Epoch: 7,training loss: 0.13 , validation loss: 1.40 , validation intent detection accuracy: 0.47
100% 720/720 [18:01<00:00, 1.34s/it]
Epoch: 8,training loss: 0.11 , validation loss: 1.35 , validation intent detection accuracy: 0.51
100% 720/720 [18:02<00:00, 1.34s/it]
Epoch: 9,training loss: 0.09 , validation loss: 1.07 , validation intent detection accuracy: 0.61
```

اینکه در هر ایپاک دقت اینتنت بیشتر میشود و loss در هر ایپاک کم و کمتر میشود به معنی این است که مدل Intent classification در حال یادگیری است. با توجه به وقت و سخت افزار در دسترس این عمل یادگیری طی فقط ده ایپاک انجام شده است و به دقت ۶۱ رسیده ایم که مشخصاست با ایپاک های بیشتر به دقت بالاتر نیز خواهد رسید.

در slot filing که میزان f1ش در micro avg وweighted avg ، ۹۲ درصد شده است که در زیر مشاهده میشود:

	Precision	recall	f1-score	support
artist_name	0.82	0.68	0.74	80
playlist_name	0.71	0.12	0.20	43
transport_agency	0.73	0.89	0.80	9
movie_type	0.00	0.00	0.00	3
house_place	0.98	0.55	0.70	73
change_amount	1.00	0.57	0.73	21
event_name	0.82	0.63	0.71	416
Other	0.94	0.97	0.96	17637
coffee_type	1.00	0.56	0.71	9
meal_type	0.64	1.00	0.78	18
currency_name	0.86	0.97	0.91	67
time_zone	0.91	0.32	0.48	31
transport_type	0.94	0.80	0.86	64
device_type	0.94	0.74	0.83	119
personal_info	0.75	0.89	0.81	27
list_name	0.80	0.55	0.65	92
podcast_descriptor	0.50	0.03	0.06	67
game_name	0.98	0.79	0.88	53
food_type	0.80	0.72	0.76	116
sport_type	0.00	0.00	0.00	0

general_frequency	0.82	0.82	0.82	45
transport_name	0.00	0.00	0.00	16
app_name	0.57	0.40	0.47	10
ingredient	0.00	0.00	0.00	10
media_type	0.78	0.91	0.84	253
joke_type	1.00	0.40	0.57	15
audiobook_author	1.00	0.14	0.25	7
business_type	0.67	0.29	0.41	48
color_type	1.00	0.81	0.89	31
weather_descriptor	0.86	0.46	0.60	125
player_setting	0.76	0.32	0.45	90
time	0.87	0.78	0.82	422
timeofday	0.92	0.78	0.84	73
music_album	0.00	0.00	0.00	1
transport_descriptor	0.00	0.00	0.00	4
order_type	0.61	0.45	0.52	31
date	0.86	0.92	0.89	622
music_descriptor	0.00	0.00	0.00	13
music_genre	0.86	0.48	0.62	62
radio_name	0.68	0.57	0.62	77
cooking_type	0.00	0.00	0.00	14
person	0.79	0.96	0.86	289
business_name	0.58	0.90	0.70	183
podcast_name	0.46	0.63	0.53	27
drink_type	0.00	0.00	0.00	1
definition_word	0.81	0.80	0.80	79
email_folder	0.64	1.00	0.78	9
place_name	0.82	0.79	0.80	359
audiobook_name	0.95	0.43	0.60	46
email_address	0.95	0.95	0.95	42
song_name	0.52	0.21	0.30	52
relation	0.66	0.88	0.75	67
news_topic	0.89	0.24	0.38	104
alarm_type	0.00	0.00	0.00	4
game_type	0.00	0.00	0.00	0
movie_name	0.00	0.00	0.00	5
pad	0.00	0.00	0.00	0
micro avg	0.92	0.92	0.92	22181
macro avg	0.62	0.49	0.52	22181
weighted avg	0.92	0.92	0.91	22181

منابع:

https://github.com/alexa/massive

https://shreelakshmigp1995.medium.com/bert-for-joint-intent-classification-and-slot-filling-1baf32e27386

https://arxiv.org/pdf/1902.10909.pdf