



بنام خدا

دانشکده‌ی مهندسی برق و کامپیوتر

درس پردازش زبان‌های طبیعی

شایان واصف احمدزاده ، 810197603

امیرحسین دبیری اقدم ، 810197502

تمرین کامپیوتری ۶

روز آپلود : 3 تیر

QA & NLU

اساتید : دکتر فیلی و دکتر یعقوب زاده

Table of Contents

PART 1 - Question Answering	2
Introduction	2
Dataset	2
Method	3
Tokenizer	3
ParsBERT Tokenizer	3
PersianALBERT Tokenizer	4
PreProcessing	5
max_length = 128 , doc_stride = 64	5
return_overflowing_tokens = True	6
return_offset_mapping = True	7
sequence_ids	8
max_length = 256 , doc_stride = 128	9
Features setups	10
Training	11
Evaluation	11
Results	13

Persian_QA,ParsBERT	13
Persian_QA,PersianALBERT	15
PQuAD, ParsBERT	18
PQuAD, PersianALBERT	20
ParSQuAD,ParsBERT	23
ParSQuAD,PersianALBERT	26
(PQuAD + PersianQA), ParsBERT	29
(PQuAD + PersianQA), ALBERTPersian	32
Conclusion	34
References	35
PART 2 - Natural Language Understanding	36
Abstract	36
Dataset	36
Method	41
Results	42
mT5 based model	42
XLM-RoBERTa based model	46
Conclusion	51
References	51
Appendix	52

*نوت‌بوک‌های مربوط به هر بخش در کنار این گزارش ضمیمه شده‌اند؛ علاوه بر آن در این [لینک](#) می‌توان از طریق گوگل درایو به نوت‌بوک‌ها و ... دسترسی پیدا کرد.

PART 1 - Question Answering

Introduction

در ابتدا به این بپردازیم که سیستم های QA قرار است چه کاری انجام دهند ؟ کاربران عموماً سوالات بخصوصی دارند که امیدوارند که یک مرجع خاص قادر به پاسخگویی به آنها باشد. این مرجع می تواند چند دست نوشته، صفحات وب مثل ویکی پدیا یا یک سری اطلاعات ثبت شده در یک Database باشد. مشکل این Task از نگاه کاربر، پیدا کردن جواب مناسب می باشد. سیستم های QA یک سوال را به عنوان ورودی دریافت می کنند و یک یا چند پاسخ را بر اساس اولویت برمی گردانند. در نهایت ما باید بتوانیم که پاسخ داده شده توسط مدل را از منظر دقت بررسی کنیم، در نتیجه نیاز است تعدادی معیار یا به اصطلاح متریک را تعریف کنیم چون بررسی وضعیت پاسخ داده شده توسط مدل توسط انسان بسیار هزینه بر است و همچنین امتیاز داده شده توسط انسان به یک سیستم QA می تواند فرد به فرد متفاوت باشد و باید از یک معیار واحد استفاده کرد. همچنین هر چقدر سیستم QA ما بزرگتر باشد باید دسته های متنوع تری از سوالات را پوشش دهیم تا مدل بتواند عملکرد مناسبی داشته باشد. در پارت اول این پروژه به بررسی چند مدل مبتنی بر ترنسفورمر QA برای چند دیتاست فارسی پرداخته و با معیارهایی که در ادامه معرفی خواهد شد آنها را ارزیابی می کنیم.

Dataset

طبق گفته سوال ما در مجموع از سه دیتاست فارسی 'persian_QA'، 'PQuAD_public' و 'ParSQuAD' استفاده می کنیم که دیتاست های مشابه SQuAD که یکی از معروف ترین دیتاست ها برای QA است می باشند. دیتاست های 'persian_QA' و 'ParSQuAD' تنها دارای دو بخش آموزش و ارزیابی هستند درحالی که دیتاست 'PQuAD' دارای سه بخش آموزش، ارزیابی و تست می باشد و بنابراین برای ارزیابی هر یک از حالت، از دادگان تست موجود در 'PQuAD' استفاده می کنیم.

از نظر حجم بودن دیتاست، دیتاست های 'pquad' و 'persian_QA' دارای حدوداً 9000 نمونه در بخش آموزش می باشند درحالی که دیتاست 'ParSQuAD' دارای دو حالت مختلف 'manual' و 'automatic' می باشد که تعداد دادگان آموزش در حالت 'manual' برابر 16000 نمونه و در حالت 'automatic' برابر 64000 نمونه می باشد.

به دلیل اینکه آموزش مدل در حالت 'automatic' حدوداً 3 ساعت طول می کشید از دادگان حالت 'manual' استفاده کردیم. برای خواندن دیتاست های بحث شده، باید به کمک کتابخانه 'load_dataset' در 'hugging face'، دیتاست های داده شده را 'load' کنیم. برای دو دیتاست 'PQuAD_public' و 'ParSQuAD' که در 'hugging face' از قبل پیاده سازی نشده بود، تنظیمات لازم را انجام دادیم که بتوانیم به کمک 'hugging face' آنها را بخوانیم :

```
dataset = load_dataset("SajjadAyoubi/persian_qa")
dataset_eval = load_dataset("Shayanvsf/pquad_public")
```

نوع داده خروجی دستور load_dataset از نوع DatasetDict می‌باشد که می‌توان به طور مستقیم از آن در مدل‌های Transformer موجود در hugging face استفاده کرد :

```
DatasetDict({
  train: Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 9008
  })
  validation: Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 930
  })
})
```

همانطور که مشاهده می‌شود، هر دیتاست دارای ستون‌های 'id'، 'title'، 'context'، 'question' و 'answer' می‌باشد که 'context' در واقع متنی هست که پاسخ (answer) یک پرسش (question) در آن وجود دارد. در زیر خلاصه دیتاست مدل برای دادگان آموزش آورده شده است :

id	title	context	question	answers
0	1	شرکت فولاد مبارکه اصفهان ...	شرکت فولاد مبارکه در کجا واقع شده است	{'text': ['در شرق شهر مبارکه'], 'answer_start': ...}
1	2	شرکت فولاد مبارکه اصفهان ...	فولاد مبارکه چند بار برنده جایزه شرکت دانشی را	{'text': ['۶'], 'answer_start': [263]}
2	3	شرکت فولاد مبارکه اصفهان ...	شرکت فولاد مبارکه در سال ۱۳۹۱ چه جایزه ای برد؟	{'text': ['تعالی سازمانی'], 'answer_start': ...}
3	4	شرکت فولاد مبارکه اصفهان ...	بزرگ ترین مجموعه تولید فولاد ایران چیست؟	{'text': ['شرکت فولاد مبارکه'], 'answer_start': ...}
4	5	شرکت فولاد مبارکه اصفهان ...	فولاد مبارکه در چه سالی احداث شد؟	{'text': ['۱۳۷۱'], 'answer_start': [504]}

Fig1. Dataset summary

Method

Tokenizer

ParsBERT Tokenizer

در زیر اطلاعات مربوط به مدل ParsBERT و همچنین نحوه Tokenize آن آورده شده است :

```
Model config BertConfig {
  "_name_or_path": "HooshvareLab/bert-base-parsbert-uncased",
  "architectures": [
    "BertForMaskedLM"
  ],
```

```

"attention_probs_dropout_prob": 0.1,
"classifier_dropout": null,
"hidden_act": "gelu",
"hidden_dropout_prob": 0.1,
"hidden_size": 768,
"initializer_range": 0.02,
"intermediate_size": 3072,
"layer_norm_eps": 1e-12,
"max_position_embeddings": 512,
"model_type": "bert",
"num_attention_heads": 12,
"num_hidden_layers": 12,
"pad_token_id": 0,
"position_embedding_type": "absolute",
"transformers_version": "4.20.0",
"type_vocab_size": 2,
"use_cache": true,
"vocab_size": 100000
}

```

در QA، ورودی Tokenizer اول سوال و بعد متن مربوطه می‌باشد :

tokenizer("اسم من شایان است", "؟ اسمتون چی بود.")

```

{'input_ids': [2, 11, 7, 36254, 4781, 4322, 46, 2125, 3, 2415, 111, 3800, 20, 3],
'token_type_ids': [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1],
'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]}

```

PersianALBERT Tokenizer

در زیر اطلاعات مربوط به مدل PersianALBERT و همچنین نحوه Tokenize آن آورده شده است :

```

Model config AlbertConfig {
  "_name_or_path": "m3hrdadfi/albert-fa-base-v2",
  "architectures": [
    "AlbertForMaskedLM"
  ],
  "attention_probs_dropout_prob": 0,
  "bos_token_id": 2,
  "classifier_dropout_prob": 0.1,
  "down_scale_factor": 1,
  "embedding_size": 128,
  "eos_token_id": 3,
  "gap_size": 0,
  "hidden_act": "gelu_new",
  "hidden_dropout_prob": 0,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "inner_group_num": 1,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,

```

```

"max_position_embeddings": 512,
"model_type": "albert",
"net_structure_type": 0,
"num_attention_heads": 12,
"num_hidden_groups": 1,
"num_hidden_layers": 12,
"num_memory_blocks": 0,
"pad_token_id": 0,
"position_embedding_type": "absolute",
"transformers_version": "4.20.1",
"type_vocab_size": 2,
"vocab_size": 80000
}

```

tokenizer("اسم من شایان است", "؟ اسمتون چی بود.")

```
{'input_ids': [2, 11, 7, 36254, 4781, 4322, 46, 2125, 3, 2415, 111, 3800, 20, 3],
```

```
'token_type_ids': [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1], 'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]}
```

PreProcessing

پارامترهایی که در طول آموزش هر 8 حالت مختلف استفاده می‌کنیم به صورت زیر می‌باشد :

```
{max_length = 256 , doc_stride = 128 ,batch_size = 8 ,lr = 3e-5 ,epoch = 3}
```

برای اینکه بیشتر با نحوه کارکرد دو پارامتر max_length و doc_stride آشنا شویم، به ترتیب دو حالت مختلف را بررسی می‌کنیم :

max_length = 128 , doc_stride = 64

(**فرض کنید از ParsBERT به عنوان Tokenizer استفاده کردیم)

با بررسی هر سطر از دیتاست آموزش، اولین سطر که تعداد token های سوال (question) به همراه متن (context) آن بزرگتر از 128 باشد، به صورت زیر می‌باشد که طولی برابر 190 دارد :

```
{'answers': {'answer_start': [114], 'text': 'در شرق شهر مبارکه'},
'context':
```

'شرکت فولاد مبارکه اصفهان، بزرگ‌ترین واحد صنعتی خصوصی در ایران و بزرگ‌ترین مجتمع تولید فولاد در کشور ایران است، که در شرق شهر مبارکه قرار دارد. فولاد مبارکه هم‌اکنون محرک بسیاری از صنایع بالادستی و پایین‌دستی است. فولاد مبارکه در ۱۱ دوره جایزه ملی تعالی سازمانی و ۶ دوره جایزه شرکت دانشی در کشور رتبه نخست را بدست آورده‌است و همچنین این شرکت در سال ۱۳۹۱ برای نخستین‌بار به عنوان تنها شرکت ایرانی با کسب

امتیاز ۶۵۴ تندیس زرین جایزه ملی تعالی سازمانی را از آن خود کند. شرکت فولاد مبارکه اصفهان در ۲۳ دی ماه ۱۳۷۱ احداث شد و اکنون بزرگ‌ترین واحدهای صنعتی و بزرگترین مجتمع تولید فولاد در ایران است. این شرکت در زمینی به مساحت ۳۵ کیلومتر مربع در نزدیکی شهر مبارکه و در ۷۵ کیلومتری جنوب غربی شهر اصفهان واقع شده است. مصرف آب این کارخانه در کمترین میزان خود، ۱۰۵٪ از دبی زاینده رود برابر سالانه ۲۳ میلیون متر مکعب در سال است و خود یکی از عوامل کم‌آبی زاینده رود شناخته می‌شود.

'id': 1,

'question': 'شرکت فولاد مبارکه در کجا واقع شده است',

'title': 'شرکت فولاد مبارکه اصفهان'

return_overflowing_tokens = True

برای آنکه token هایی که با یکدیگر همپوشانی دارند، در قالبی لیستی برگردانده شوند، نیاز است که پارامتر return_overflowing_tokens را برابر True قرار دهیم :

```
tokenized_example = tokenizer( example["question"], example["context"], max_length = max_length,
truncation="only_second", return_overflowing_tokens=True, stride=doc_stride )
```

حال به کمک دستور زیر، طول token های بدست آمده در هر دسته اختصاص داده شده را می‌بینیم :

```
[len(x) for x in tokenized_example["input_ids"]]
```

```
[128, 128, 84]
```

در زیر token های آمده در هر دسته را با هم بررسی می‌کنیم :

[CLS] شرکت فولاد مبارکه در کجا واقع شده است [SEP] شرکت فولاد مبارکه اصفهان ، بزرگترین واحد صنعتی خصوصی در ایران و بزرگترین مجتمع تولید فولاد در کشور ایران است ، که در شرق شهر مبارکه قرار دارد. فولاد مبارکه هم‌اکنون محرک بسیاری از صنایع بالادستی و پاییندستی است. فولاد مبارکه در ۱۱ دوره جایزه ملی تعالی سازمانی و ۶ دوره جایزه شرکت دانشی در کشور رتبه نخست را بدست آورده‌است و همچنین این شرکت در سال ۱۳۹۱ برای نخستین‌بار به عنوان تنها شرکت ایرانی با کسب امتیاز ۶۵۴ تندیس زرین جایزه ملی تعالی سازمانی را از آن خود کند. شرکت فولاد مبارکه اصفهان در ۲۳ دی ماه ۱۳۷۱ احداث شد و اکنون بزرگترین واحدهای صنعتی و بزرگترین مجتمع تولید فولاد در [SEP]

[CLS] شرکت فولاد مبارکه در کجا واقع شده است [SEP] دوره جایزه شرکت دانشی در کشور رتبه نخست را بدست آورده‌است و همچنین این شرکت در سال ۱۳۹۱ برای نخستین‌بار به عنوان تنها شرکت ایرانی با کسب امتیاز ۶۵۴ تندیس زرین جایزه ملی تعالی سازمانی را از آن خود کند. شرکت فولاد مبارکه اصفهان در ۲۳ دی ماه ۱۳۷۱ احداث شد و اکنون بزرگترین واحدهای صنعتی و بزرگترین مجتمع تولید فولاد در ایران است. این شرکت در زمینی به مساحت ۳۵ کیلومتر مربع در نزدیکی شهر مبارکه و در ۷۵ کیلومتری جنوب غربی شهر اصفهان واقع شده‌است. مصرف آب این کارخانه در کمترین میزان خود ، ۱۰۵٪ از دبی زاینده‌رود برابر سالانه ۲۳ میلیون متر مکعب در سال است و [SEP]

[[CLS]] شرکت فولاد مبارکه در کجا واقع شده است [SEP] و اکنون بزرگترین واحدهای صنعتی و بزرگترین مجتمع تولید فولاد در ایران است. این شرکت در زمینی به مساحت ۳۵ کیلومتر مربع در نزدیکی شهر مبارکه و در ۷۵ کیلومتری جنوب غربی شهر اصفهان واقع شده‌است. مصرف آب این کارخانه در کمترین میزان خود، ۱،۵٪ از دبی زاینده‌رود برابر سالانه ۲۳ میلیون متر مکعب در سال است و خود یکی از عوامل کمابی زاینده‌رود شناخته می‌شود.

[SEP]

- هر کدام از دسته‌ها با سوال (question) شروع می‌شود که با رنگ آبی مشخص شده‌اند و در ادامه متن (context) می‌آید.
- Token های خاص مربوطه در شروع هر دسته با CLS، مرز بین سوال با متن با SEP و همچنین پایان متن ناقص در هر دست با SEP نشان داده می‌شوند.
- در هر دسته، متن مشترک آن با دسته مجاور بعدی مشخص شده است (دسته اول با دوم با رنگ قرمز، دسته دوم با سوم با _ مشخص شده‌اند) که طول هر یک برابر token 64 می‌باشد. همچنین در دسته آخر، token 84 باقی‌مانده است که نشان می‌دهد تعداد Token های بخش نارنجی برابر token 20 می‌باشد.
- همانطور که مشخص است، دو دسته اول هر کدام بیشترین مقدار مجازی که توسط max_length تعیین شده (در اینجا 128) را دارند و درایه آخر token های آخر باقی مانده را تشکیل می‌دهد که طبیعتاً از 128 کمتر هست.
- پس همانطور که مشاهده شد، تعداد ویژگی‌های بدست آمده از هر مثال با پارامتری بنام doc_stride کنترل می‌شود. در واقع برای یک max_length مشخص، doc_stride های مختلف باعث ایجاد تعداد ویژگی‌های متفاوتی برای هر مثال می‌شوند.
- نکته دیگر آنکه اگر max_length از طول بیشترین جمله موجود در دیتاست بیشتر باشد، آنگاه دیگر پارامتر doc_stride تاثیری در تعداد ویژگی‌های تولید شده ندارد و همواره یک عنصر (یک ویژگی) برگردانده می‌شود.

return_offset_mapping = True

مدل ما در نهایت برای پیشبینی پاسخ یک سوال دو اندیس شروع و خاتمه برمیگرداند که به معنی این است که پاسخ سوال از Token شماره چند شروع شده و به Token شماره چند ختم می‌شود، نمونه ای از خروجی مدل بعد از آموزش به صورت زیر است:

```
output.keys()
odict_keys(['loss', 'start_logits', 'end_logits'])
```

Fig2. Model output's keys

همچنین در دیتاست، برای هر سطر قسمتی بنام start_answer مشخص شده است که بیان می‌کند پاسخ سوال موردنظر از چه کاراکتری شروع می‌شود.

از این دو نتیجه می‌گیریم که به مفهومی نیاز داریم که بتواند اندیس کاراکترهای موجود در Token های درون متن و سوال را برگرداند که به آن offset_mapping می‌گوییم.

در زیر، برای هر Token یک tuple به شکل (start , end) برگردانده می‌شود که نشان می‌دهد هر token از کاراکتر شماره چند شروع و به کاراکتر شماره چند ختم می‌شود. برای مثال برای متن موجود در دسته اول که به صورت زیر است، تا 20 خروجی اول را چاپ می‌کنیم :

[CLS] شرکت فولاد مبارکه در کجا واقع شده است [SEP] شرکت فولاد مبارکه اصفهان ، بزرگترین واحد صنعتی خصوصی در ایران و بزرگترین مجتمع تولید فولاد در کشور ایران است ، که در شرق شهر مبارکه قرار دارد. فولاد مبارکه همانکون محرک بسیاری از صنایع بالادستی و پایبندستی است. فولاد مبارکه در ۱۱ دوره جایزه ملی تعالی سازمانی و ۶ دوره جایزه شرکت دانشی در کشور رتبه نخست را بدست آوردهاست و همچنین این شرکت در سال ۱۳۹۱ برای نخستینبار به عنوان تنها شرکت ایرانی با کسب امتیاز ۶۵۴ تندیس زرین جایزه ملی تعالی سازمانی را از ان خود کند. شرکت فولاد مبارکه اصفهان در ۲۳ دی ماه ۱۳۷۱ احداث شد و اکنون بزرگترین واحدهای صنعتی و بزرگترین مجتمع تولید فولاد در [SEP]

```
tokenized_example = tokenizer(example["question"], example["context"], max_length=max_length,
                               truncation="only_second", return_overflowing_tokens=True, return_offsets_mapping=True,
                               stride=doc_stride)

print(tokenized_example["offset_mapping"][0][:20])
```

[(0, 0), (0, 4), (5, 10), (11, 17), (18, 20), (21, 24), (25, 29), (30, 33), (34, 37), (0, 0), (0, 4), (5, 10), (11, 16), (16, 17), (18, 24), (24, 25), (26, 35), (36, 40), (41, 46), (47, 52)]

همانطور که ذکر شد، اولین Token در هر دسته CLS می‌باشد که در واقع یک شروع و خاتمه ندارد بنابراین (0,0) برگردانده می‌شود. سپس Token بعدی کلمه شرکت می‌باشد که دارای 4 کاراکتر می‌باشد پس (0,4) برگردانده می‌شود و همینطور برای بقیه Token ها.

sequence_ids

در نهایت برای اینکه تمایز میان سوال و متن را به مدل بفهمانیم از `sequenc_id` استفاده می‌کنیم به طوریکه به Token های خاص مقدار `None` به Token های مربوط به سوال، مقدار `0` و به Token های مربوط به متن، مقادیر `1` را برمی‌گرداند :

[illegible]

max_length = 256 , doc_stride = 128

(**فرض کنید از PersianALBERTA به عنوان Tokenizer استفاده کردیم)

با بررسی هر سطر از دیتاست آموزش، اولین سطر که تعداد token های سوال (question) به همراه متن (context) آن بزرگتر از 256 باشد، به صورت زیر می باشد که طولی برابر 258 دارد :

```
{'answers': {'answer_start': [0], 'text': 'بستنی'}}
```

'بستنی' نوعی دسر منجمد و مزه \u200c\u200c است که باید شامل حداقل ۱۰٪ چربی شیر باشد. مقدار این چربی می \u200c\u200c تواند از ۱۰ تا ۱۶ درصد تغییر یابد که معمولاً بستنی \u200c\u200c ها حاوی ۱۴٪ چربی شیر هستند، همچنین بستنی می \u200c\u200c تواند شامل دیگر انواع لبنیات مثل خامه باشد که به همراه مواد شیرین \u200c\u200c کننده و طعم \u200c\u200c دهنده تهیه می \u200c\u200c شود. اولین کارخانه صنعتی بستنی \u200c\u200c سازی جهان در سال ۱۸۵۱ در مریلند، کار خود را آغاز نمود. بستنی معروف ترین دسر در جهان است. قدمت دسرهای یخی در چین به ۴ هزار سال قبل می \u200c\u200c رسد. یک نوع از این دسر ها با قرار دادن سنگ نمک و برف در جداره \u200c\u200c های ظروف حاوی شربت تهیه می \u200c\u200c شد (نمک باعث می \u200c\u200c شود دمای یخ \u200c\u200c زدن آب به زیر صفر برسد) و نوع دیگری با قرار دادن شیر، برنج کاملاً پخته به همراه ادویه در داخل برف تهیه می \u200c\u200c شد. بستنی یخی هم با آب \u200c\u200c میوه، عسل و ادویه درست می \u200c\u200c شد. این دسر های یخی از مسیرهای تجاری وارد ایران هم شدند. همچنین حدود ۴ هزار سال پیش در ایران دسر یخی فالوده با استفاده از یخ، گلاب، آرد برنج رشته \u200c\u200c ای و زعفران و طعم \u200c\u200c دهنده \u200c\u200c های دیگر ساخته می \u200c\u200c شد. نوعی بستنی با نام بستنی سنتی یا ایرانی در ایران رواج دارد که دارای طعم زعفران و رنگ زرد می \u200c\u200c باشد؛

'id': 292,

'question': 'معروف ترین دسر در دنیا کدام است؟'

'title': 'بستنی'

در زیر token های آمده در هر دسته را با هم بررسی می کنیم :

```
[CLS] معروف ترین دسر در دنیا کدام است؟ [SEP] بستنی نوعی دسر منجمد و مزه دار است که باید شامل حداقل ۱۰<unk> چربی شیر باشد. مقدار این چربی می تواند از ۱۰ تا ۱۶ درصد تغییر یابد که معمولاً بستنی ها حاوی ۱۴<unk> چربی شیر هستند، همچنین بستنی می تواند شامل دیگر انواع لبنیات مثل خامه باشد که به همراه مواد شیرین کننده و طعم دهنده تهیه می شود. اولین کارخانه صنعتی بستنی سازی جهان در سال ۱۸۵۱ در مریلند، کار خود را آغاز نمود. بستنی معروف ترین دسر در جهان است. قدمت دسرهای یخی در چین به ۴ هزار سال قبل می رسد. یک نوع از این دسر ها با قرار دادن سنگ نمک و برف در جداره های ظروف حاوی شربت تهیه می شد (نمک باعث می شود دمای یخ زدن اب به زیر صفر برسد) و نوع دیگری با قرار دادن شیر، برنج کاملاً پخته به همراه ادویه در داخل برف تهیه می شد. بستنی یخی هم با اب میوه، عسل و ادویه درست می شد. این دسر های یخی از مسیرهای تجاری وارد ایران هم شدند. همچنین حدود ۴ هزار سال پیش در ایران دسر یخی فالوده با استفاده از یخ، گلاب،
```

ارد برنج رشته ای و زعفران و طعم دهنده های دیگر ساخته می شد. نوعی بستنی با نام بستنی سنتی یا ایرانی در ایران رواج دارد که دارای طعم زعفران و رنگ زرد [SEP]

[CLS] معروف ترین دسر در دنیا کدام است؟ [SEP] در جداره های ظروف حاوی شربت تهیه می شد (نمک باعث می شود دمای یخ زدن آب به زیر صفر برسد) و نوع دیگری با قرار دادن شیر، برنج کاملاً پخته به همراه ادویه در داخل برف تهیه می شد. بستنی یخی هم با آب میوه، عسل و ادویه درست می شد. این دسرهای یخی از مسیرهای تجاری وارد ایران هم شدند. همچنین حدود ۴ هزار سال پیش در ایران دسر یخی فالوده با استفاده از یخ، گلاب، ارد برنج رشته ای و زعفران و طعم دهنده های دیگر ساخته می شد. نوعی بستنی با نام بستنی سنتی یا ایرانی در ایران رواج دارد که دارای طعم زعفران و رنگ زرد می باشد [SEP]

همچنین خروجی offset_mapping به صورت زیر می باشد :

[(0, 0), (0, 5), (5, 10), (10, 14), (14, 17), (17, 22), (22, 27), (27, 31), (31, 32), (0, 0), (0, 5), (5, 10), (10, 14), (14, 20), (20, 22), (22, 26), (26, 30), (30, 34), (34, 37), (37, 42), (42, 47)]

در اینجا برای مثال (0,5) مربوط به Token "معروف" می باشد.

Features setups

حال که با سه مفهوم overflow, offset و sequence_ids آشنا شدیم، نیاز به آماده سازی تابعی می باشد که بر روی تمامی مثال های دیتاست اعمال شود. در دیتاست داده شده، پاسخ یک سوال و اندیس شروع آن داده شده است ولی سوال این است اگر هر مثال به چند دسته تقسیم می شود و به اصلاح دارای چند ویژگی است، چگونه پاسخ اولیه داده شده را هنگامی که متن اصلی ما به چند دسته تقسیم شده پیدا کنیم ؟

در مثال ذکر شده در بالا هنگامی که برای یک مثال سه دسته ویژگی مختلف بدست می آوریم، امکان دارد پاسخ مربوطه در هر کدام از دسته های بوجود آمده باشد. به این منظور تابعی به اسم Prepare_Train_Features ایجاد می کنیم که در ابتدا اندیس های شروع و پایان پاسخ در متن اصلی را بگیرد و سپس دو اندیس محلی start_index و end_index که اندیس های مربوط به یک token می باشند را تعریف می کند تا بین ویژگی های مختلف عملیات search را انجام دهد.

در نهایت به کمک دستور map در کتابخانه hugging face، تابع نوشته شده را بر تمامی مثال ها اعمال می کنیم :

```
tokenized_ds = dataset.map(prepare_train_features, batched=True,  
                           remove_columns=dataset["train"].column_names)
```

همین کار را برای دادگان تست بعد از پیشبینی مدل انجام می دهیم تا بتوانیم از اندیس های پیش بینی شده، بهترین متن پاسخ خروجی را برگردانیم.

Training

حال که تمامی مراحل لازم را توضیح دادیم، نوبت به آموزش مدل می‌رسد :

```
model_name = model_checkpoint.split("/")[-1]

args = TrainingArguments(
    output_dir = f"/content/drive/MyDrive/CA6/checkpoints/{model_name}-finetuned-PersianQA",
    overwrite_output_dir = True,
    save_strategy = "steps",
    save_steps = 1000,
    evaluation_strategy = "epoch",
    learning_rate=lr,
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    num_train_epochs=epoch,
    weight_decay=0.0001)
```

تعدادی از پارامترهای بالا طبق پارامترهای بهینه در مدل آموزش داده شده بر روی دیتاست Squad قرار داده شده‌اند. برای آنکه در هر 1000 پله از آموزش مدل، وزن‌های مربوطه را ذخیره کنیم، save_strategy را بر روی 'steps' قرار داده و save_steps را برابر 1000 قرار می‌دهیم.

Evaluation

همانطور که ذکر شد، خروجی مدل شامل مقادیر خطا، اندیس اول و اندیس دوم می‌باشد. برای مثال خروجی مدل برای دادگان ارزیابی برای یک batch از دیتا به صورت زیر می‌باشد :

```
putput.start_logits.shape, output.end_logits.shape
(torch.Size([8, 128]), torch.Size([8, 128]))
```

Fig3. Model logits's shapes

که اندیس دوم همان تعداد Token ها در هر بسته می‌باشد که در اینجا چون doc_stride برابر 128 هست، بیشتر از 128 نمی‌شود.

در فرآیند بازیابی پاسخ از خروجی مدل، دو شرط محدودکننده که در واقع دو پارامتر می‌باشند را تعیین می‌کنیم :

1. **best_n**: به جای آنکه از بین هر 128 (یا کمتر از 128) Token برگردانده شده، بیشترین احتمال را به عنوان Token محتمل مدل برگردانیم، از **n_best** بهترین Token هایی که مدل پیشبینی کرده انتخاب می‌کنیم. در واقع در ابتدا Token های موجود در هر مثال را بر حسب احتمال مرتب می‌کنیم و **n_best** تا از اول را انتخاب می‌کنیم.

2. **max_length**: در هنگامی که یک متن به عنوان پاسخ برگردانده می‌شود ابتدا طول آنرا محاسبه می‌کنیم و در صورتی که تعداد Token های موجود در پاسخ از **max_length** بیشتر باشد، آنرا حذف می‌کنیم.

در نهایت نیاز به برای مقایسه مقادیر پیشبینی شده با واقعی از متریک 'squad_v2' استفاده می‌کنیم که برای هر دو ورودی مرجع و پیش‌بینی شده، سه مقدار را باید مشخص کرد :

1. **'id'**: طبیعتاً برای پیدا کردن پاسخ پیشبینی شده متناظر با پاسخ مرجع، نیاز به 'id' هر جمله در دادگان تست داریم.

2. **'prediction_test' / 'answers'**: که اولی به صورت یک Dict و دومی به صورت متن می‌باشد.

3. **'no answer threshold' / 'no answer probability'**: یک احتمال است که بیان می‌کند یک سوال به چه احتمالی پاسخ ندارد.

در نهایت به کمک دستور **metric.compute**، خروجی های زیر برگردانده می‌شود :


```
metric.compute(predictions=formatted_predictions, references=references)
```

1. **exact**: معیاری هست که در سطح کاراکتر، پاسخ مرجع و پیشبینی شده را مقایسه می‌کند.
2. **F1**: معیاری هست که در سطح لغت، پاسخ مرجع و پیشبینی شده را مقایسه می‌کند.
3. **total**: تعداد کل پاسخ‌ها.
4. **HasAns_exact**: معیار 'exact' برای جملاتی که دارای جواب هستند، محاسبه می‌شود.
5. **HasAns_f1**: معیار f1 برای جملاتی که دارای جواب هستند، محاسبه می‌شود.
6. **HasAns_total**: چه تعداد از سوالات دارای جواب هستند.
7. **NoAns_exact**: معیار 'exact' برای جملاتی که دارای جواب نیستند، محاسبه می‌شود.
8. **NoAns_f1**: معیار f1 برای جملاتی که دارای جواب نیستند، محاسبه می‌شود.
9. **NoAns_Total**: چه تعداد از سوالات دارای جواب نیستند.
10. **best_exact**: بهترین exact بدست آمده.
11. **best_exact_threshold**: احتمال 'no_answer_probability' برای بهترین exact.
12. **best_f1**: بهترین f1 بدست آمده.
13. **best_f1_threshold**: احتمال 'no_answer_probability' برای بهترین f1.

Results

Persian_QA,ParsBERT

بعد از epoch 3 آموزش مدل، نمودار خطای دادگان آموزش و ارزیابی به صورت زیر می باشد :



Epoch	Training Loss	Validation Loss
1	0.940400	0.933049
2	0.490200	1.083683
3	0.199900	1.700741

Fig4. Model loss for ParsBERT on Persian_QA

همچنین برای اینکه تاثیر دو پارامتر n_best و max_length را در عملکرد مدل بررسی کنیم، برای n_best از سه مقدار $[10,20,30]$ و برای max_length از سه مقدار $[10,15,20]$ استفاده می کنیم. پس در مجموع 9 حالت مختلف را بررسی می کنیم :

```
{'exact': 55.29045643153527, 'f1': 68.74102656238097, 'total': 964, 'HasAns_exact': 42.701227830832195, 'HasAns_f1': 60.390654305777964, 'HasAns_total': 733, 'NoAns_exact': 95.23809523809524, 'NoAns_f1': 95.23809523809524, 'NoAns_total': 231, 'best_exact': 55.29045643153527, 'best_exact_thresh': 0.0, 'best_f1': 68.741026562381, 'best_f1_thresh': 0.0}
```

```
{'exact': 56.95020746887967, 'f1': 70.58048777868758, 'total': 964, 'HasAns_exact': 45.020463847203274, 'HasAns_f1': 62.9462349504159, 'HasAns_total': 733, 'NoAns_exact': 94.8051948051948, 'NoAns_f1': 94.8051948051948, 'NoAns_total': 231, 'best_exact': 56.95020746887967, 'best_exact_thresh': 0.0, 'best_f1': 70.5804877786876, 'best_f1_thresh': 0.0}
```

```
{'exact': 57.572614107883815, 'f1': 72.01769144976983, 'total': 964, 'HasAns_exact': 45.839017735334245, 'HasAns_f1': 64.83636365290333, 'HasAns_total': 733, 'NoAns_exact': 94.8051948051948, 'NoAns_f1': 94.8051948051948, 'NoAns_total': 231, 'best_exact': 57.572614107883815, 'best_exact_thresh': 0.0, 'best_f1': 72.01769144976983, 'best_f1_thresh': 0.0}
```

```
{'exact': 55.29045643153527, 'f1': 68.71182306247117, 'total': 964, 'HasAns_exact': 42.701227830832195, 'HasAns_f1': 60.352247520084866, 'HasAns_total': 733, 'NoAns_exact': 95.23809523809524, 'NoAns_f1': 95.23809523809524, 'NoAns_total': 231, 'best_exact': 55.29045643153527, 'best_exact_thresh': 0.0, 'best_f1': 68.7118230624712, 'best_f1_thresh': 0.0}
```

```
{'exact': 56.95020746887967, 'f1': 70.55426184140266, 'total': 964, 'HasAns_exact': 45.020463847203274, 'HasAns_f1': 62.9117440861012, 'HasAns_total': 733, 'NoAns_exact': 94.8051948051948, 'NoAns_f1': 94.8051948051948, 'NoAns_total': 231, 'best_exact': 56.95020746887967, 'best_exact_thresh': 0.0, 'best_f1': 70.55426184140266, 'best_f1_thresh': 0.0}
```

```
{'exact': 57.572614107883815, 'f1': 72.01769144976983, 'total': 964, 'HasAns_exact': 45.839017735334245, 'HasAns_f1': 64.83636365290333, 'HasAns_total': 733, 'NoAns_exact': 94.8051948051948, 'NoAns_f1': 94.8051948051948, 'NoAns_total': 231, 'best_exact': 57.572614107883815, 'best_exact_thresh': 0.0, 'best_f1': 72.01769144976983, 'best_f1_thresh': 0.0}
```

```
{'exact': 55.29045643153527, 'f1': 68.71182306247117, 'total': 964, 'HasAns_exact': 42.701227830832195, 'HasAns_f1': 60.352247520084866, 'HasAns_total': 733, 'NoAns_exact': 95.23809523809524, 'NoAns_f1': 95.23809523809524, 'NoAns_total': 231, 'best_exact': 55.29045643153527, 'best_exact_thresh': 0.0, 'best_f1': 68.7118230624712, 'best_f1_thresh': 0.0}
```

```
{'exact': 56.95020746887967, 'f1': 70.55426184140266, 'total': 964, 'HasAns_exact': 45.020463847203274, 'HasAns_f1': 62.9117440861012, 'HasAns_total': 733, 'NoAns_exact': 94.8051948051948, 'NoAns_f1': 94.8051948051948, 'NoAns_total': 231, 'best_exact': 56.95020746887967, 'best_exact_thresh': 0.0, 'best_f1': 70.55426184140266, 'best_f1_thresh': 0.0}
```

```
{'exact': 57.572614107883815, 'f1': 72.01769144976983, 'total': 964, 'HasAns_exact': 45.839017735334245, 'HasAns_f1': 64.83636365290333, 'HasAns_total': 733, 'NoAns_exact': 94.8051948051948, 'NoAns_f1': 94.8051948051948, 'NoAns_total': 231, 'best_exact': 57.572614107883815, 'best_exact_thresh': 0.0, 'best_f1': 72.01769144976983, 'best_f1_thresh': 0.0}
```

خلاصه 9 حالت بالا برای دو خروجی 'exact' و 'f1' در جدول زیر آورده شده است :

	best_n	max_length	exact	f1
0	10.0	10.0	53.112033	67.017755
1	10.0	15.0	53.941909	68.424934
2	10.0	20.0	54.460581	70.006616
3	20.0	10.0	53.112033	67.043689
4	20.0	15.0	53.941909	68.424934

Fig5. Model performance summary for exact and f1 metrics

همچنین پیشبینی مدل را برای یک id دلخواه با مقدار مرجع مقایسه می‌کنیم :

```
[85] references[15]
{'answers': {'answer_start': [390], 'text': ['بایرن مونیخ']}, 'id': 1719487.0}

[86] formatted_predictions[15]
{'id': 1719487.0,
 'no_answer_probability': 0.0,
 'prediction_text': 'بایرن مونیخ'}
```

Fig6. Model's output comparison for prediction and reference

همچنین نتایج موجود در جدول بالا را در قالب دو نمودار مجزا برای exact و f1 در قالب scatter plot رسم کردیم :

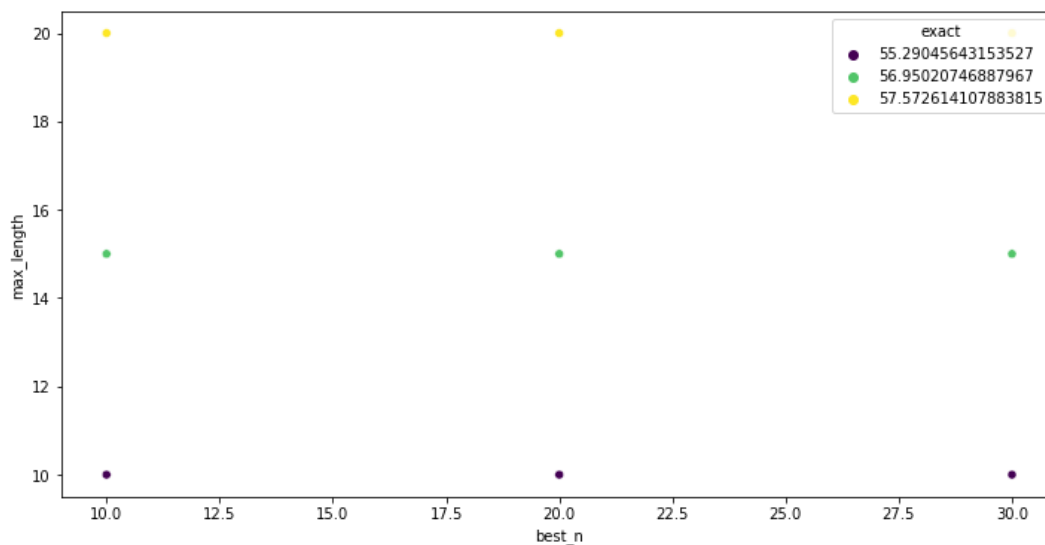


Fig7. exact metric performance for different ranges of best_n,max_length

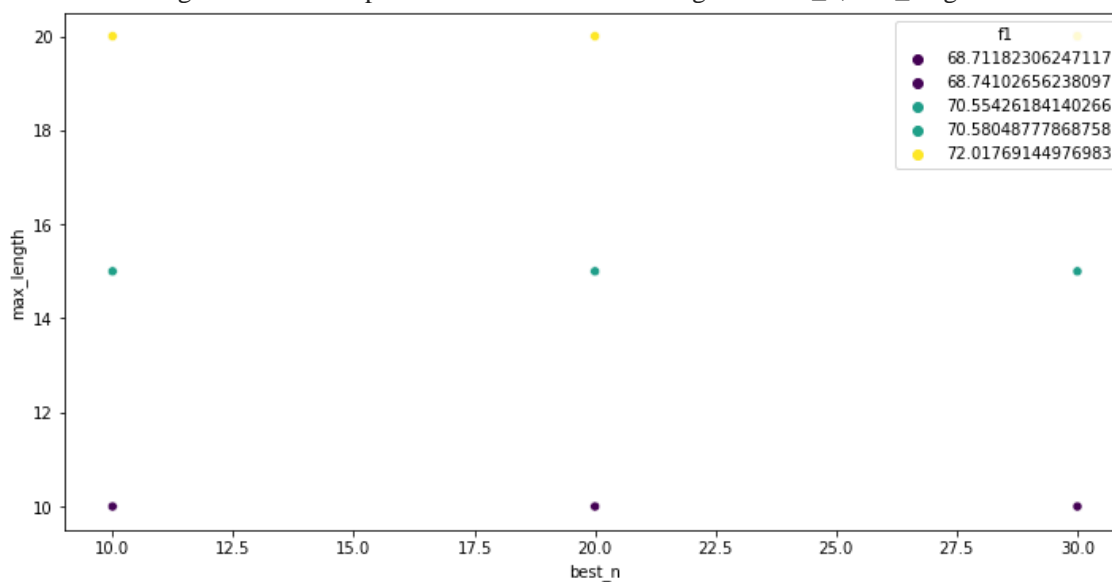


Fig8. f1 metric performance for different ranges of best_n,max_length

Persian_QA,PersianALBERT

بعد از epoch 3 آموزش مدل، نمودار خطای دادگان آموزش و ارزیابی به صورت زیر می باشد :

[4707/4707 11:11, Epoch 3/3]		
Epoch	Training Loss	Validation Loss
2	1.070400	1.450852
3	0.540600	1.741250

Fig9. Model loss for PersianALBERT on Persian_QA

همچنین برای اینکه تاثیر دو پارامتر `n_best` و `max_length` را در عملکرد مدل بررسی کنیم، برای `n_best` از سه مقدار `[10,20,30]` و برای `max_length` از سه مقدار `[10,15,20]` استفاده می‌کنیم. پس در مجموع 9 حالت مختلف را بررسی می‌کنیم:

```
{'exact': 53.63070539419087, 'f1': 67.74915965128199, 'total': 964, 'HasAns_exact': 41.06412005457026,
'HasAns_f1': 59.63190982787979, 'HasAns_total': 733, 'NoAns_exact': 93.50649350649351, 'NoAns_f1':
93.50649350649351, 'NoAns_total': 231, 'best_exact': 53.63070539419087, 'best_exact_thresh': 0.0, 'best_f1':
67.74915965128199, 'best_f1_thresh': 0.0}
```

```
{'exact': 54.66804979253112, 'f1': 70.07070485443255, 'total': 964, 'HasAns_exact': 42.42837653478854,
'HasAns_f1': 62.68507432424696, 'HasAns_total': 733, 'NoAns_exact': 93.50649350649351, 'NoAns_f1':
93.50649350649351, 'NoAns_total': 231, 'best_exact': 54.66804979253112, 'best_exact_thresh': 0.0, 'best_f1':
70.07070485443255, 'best_f1_thresh': 0.0}
```

```
{'exact': 55.49792531120332, 'f1': 71.29923196436089, 'total': 964, 'HasAns_exact': 43.519781718963166,
'HasAns_f1': 64.30076345654015, 'HasAns_total': 733, 'NoAns_exact': 93.50649350649351, 'NoAns_f1':
93.50649350649351, 'NoAns_total': 231, 'best_exact': 55.49792531120332, 'best_exact_thresh': 0.0, 'best_f1':
71.29923196436086, 'best_f1_thresh': 0.0}
```

```
{'exact': 53.63070539419087, 'f1': 67.75508733355822, 'total': 964, 'HasAns_exact': 41.06412005457026,
'HasAns_f1': 59.639705579195315, 'HasAns_total': 733, 'NoAns_exact': 93.50649350649351, 'NoAns_f1':
93.50649350649351, 'NoAns_total': 231, 'best_exact': 53.63070539419087, 'best_exact_thresh': 0.0, 'best_f1':
67.75508733355822, 'best_f1_thresh': 0.0}
```

```
{'exact': 54.66804979253112, 'f1': 70.07070485443255, 'total': 964, 'HasAns_exact': 42.42837653478854,
'HasAns_f1': 62.68507432424696, 'HasAns_total': 733, 'NoAns_exact': 93.50649350649351, 'NoAns_f1':
93.50649350649351, 'NoAns_total': 231, 'best_exact': 54.66804979253112, 'best_exact_thresh': 0.0, 'best_f1':
70.07070485443255, 'best_f1_thresh': 0.0}
```

```
{'exact': 55.49792531120332, 'f1': 71.29923196436089, 'total': 964, 'HasAns_exact': 43.519781718963166,
'HasAns_f1': 64.30076345654015, 'HasAns_total': 733, 'NoAns_exact': 93.50649350649351, 'NoAns_f1':
93.50649350649351, 'NoAns_total': 231, 'best_exact': 55.49792531120332, 'best_exact_thresh': 0.0, 'best_f1':
71.29923196436086, 'best_f1_thresh': 0.0}
```

```
{'exact': 53.63070539419087, 'f1': 67.75508733355822, 'total': 964, 'HasAns_exact': 41.06412005457026,
'HasAns_f1': 59.639705579195315, 'HasAns_total': 733, 'NoAns_exact': 93.50649350649351, 'NoAns_f1':
93.50649350649351, 'NoAns_total': 231, 'best_exact': 53.63070539419087, 'best_exact_thresh': 0.0, 'best_f1':
67.75508733355822, 'best_f1_thresh': 0.0}
```

```
{'exact': 54.66804979253112, 'f1': 70.07070485443255, 'total': 964, 'HasAns_exact': 42.42837653478854,
'HasAns_f1': 62.68507432424696, 'HasAns_total': 733, 'NoAns_exact': 93.50649350649351, 'NoAns_f1':
93.50649350649351, 'NoAns_total': 231, 'best_exact': 54.66804979253112, 'best_exact_thresh': 0.0, 'best_f1':
70.07070485443255, 'best_f1_thresh': 0.0}
```

```
{'exact': 55.49792531120332, 'f1': 71.29923196436089, 'total': 964, 'HasAns_exact': 43.519781718963166,
'HasAns_f1': 64.30076345654015, 'HasAns_total': 733, 'NoAns_exact': 93.50649350649351, 'NoAns_f1':
93.50649350649351, 'NoAns_total': 231, 'best_exact': 55.49792531120332, 'best_exact_thresh': 0.0, 'best_f1':
71.29923196436086, 'best_f1_thresh': 0.0
}
```

خلاصه 9 حالت بالا برای دو خروجی 'exact' و 'f1' در جدول زیر آورده شده است :

	best_n	max_length	exact	f1
0	10.0	10.0	53.630705	67.749160
1	10.0	15.0	54.668050	70.070705
2	10.0	20.0	55.497925	71.299232
3	20.0	10.0	53.630705	67.755087
4	20.0	15.0	54.668050	70.070705

Fig10. Model performance summary for exact and f1 metrics

همچنین پیشبینی مدل را برای یک id دلخواه با مقدار مرجع مقایسه می‌کنیم :

```
[61] references[15]

{'answers': {'answer_start': [390], 'text': ['بایرن مونیخ']}, 'id': 1719487.0}

formatted_predictions[15]

{'id': 1719487.0,
'no_answer_probability': 0.0,
'prediction_text': 'بایرن مونیخ'}
```

Fig11. Model's output comparison for prediction and reference

همچنین نتایج موجود در جدول بالا را در قالب دو نمودار مجزا برای exact و f1 در قالب scatter plot رسم کردیم :

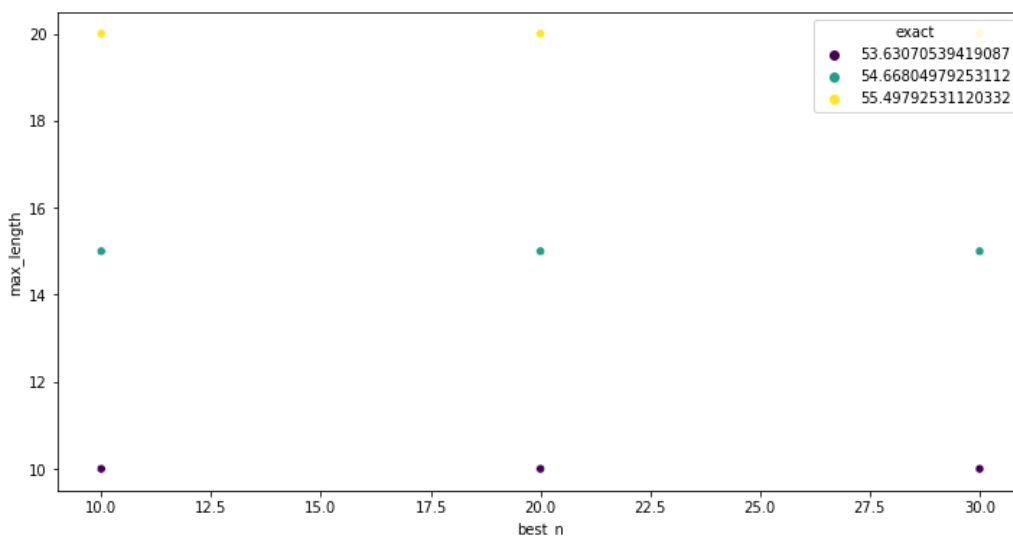


Fig12. exact metric performance for different ranges of best_n,max_length

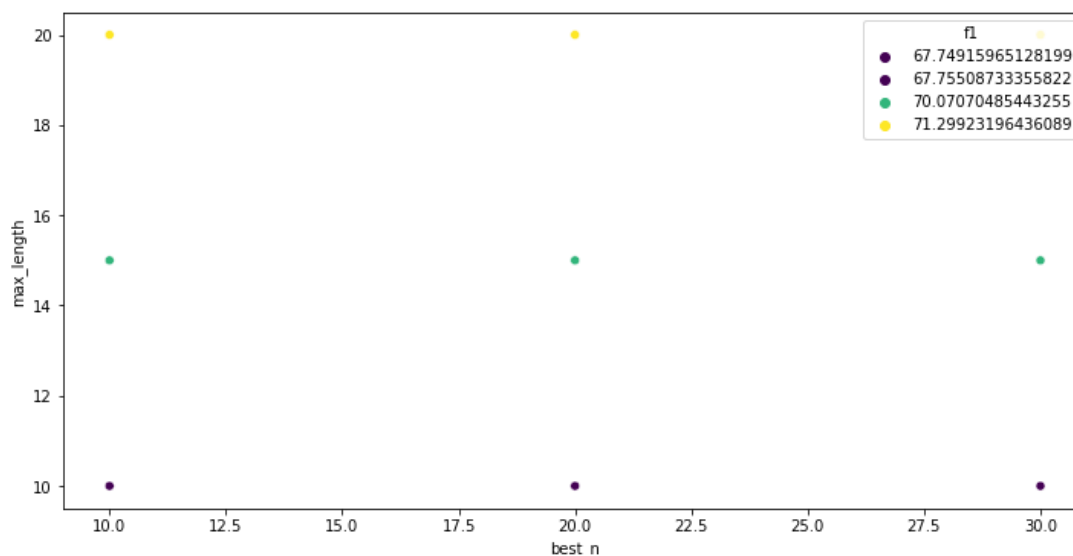


Fig13. f1 metric performance for different ranges of best_n,max_length

PQuAD, ParsBERT

بعد از epoch 3 آموزش مدل، نمودار خطای دادگان آموزش و ارزیابی به صورت زیر می باشد :

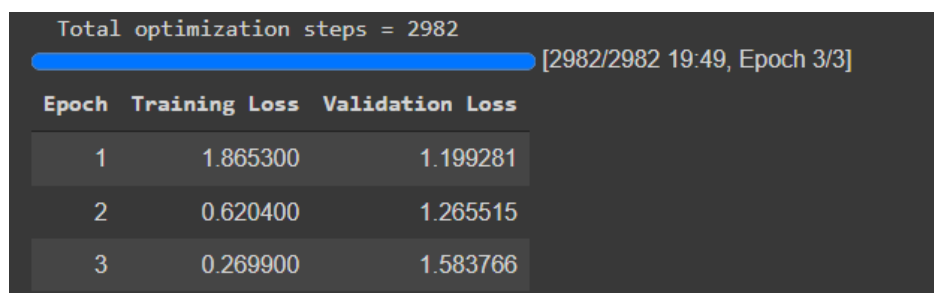


Fig14. Model loss for ParsBERT on PQuAD

همچنین برای اینکه تاثیر دو پارامتر `n_best` و `max_length` را در عملکرد مدل بررسی کنیم، برای `n_best` از سه مقدار [10,20,30] و برای `max_length` از سه مقدار [10,15,20] استفاده می کنیم. پس در مجموع 9 حالت مختلف را بررسی می کنیم :

```
{'exact': 63.27800829875519, 'f1': 76.70768674179664, 'total': 964, 'HasAns_exact': 55.38881309686221,
'HasAns_f1': 73.05076400967528, 'HasAns_total': 733, 'NoAns_exact': 88.31168831168831, 'NoAns_f1':
88.31168831168831, 'NoAns_total': 231, 'best_exact': 63.27800829875519, 'best_exact_thresh': 0.0, 'best_f1':
76.70768674179666, 'best_f1_thresh': 0.0}
```

```
{'exact': 64.93775933609959, 'f1': 78.77383855522557, 'total': 964, 'HasAns_exact': 57.70804911323329,
'HasAns_f1': 75.90447526226122, 'HasAns_total': 733, 'NoAns_exact': 87.87878787878788, 'NoAns_f1':
87.87878787878788, 'NoAns_total': 231, 'best_exact': 64.93775933609959, 'best_exact_thresh': 0.0, 'best_f1':
78.77383855522557, 'best_f1_thresh': 0.0}
```

```
{'exact': 65.35269709543569, 'f1': 80.55687172135933, 'total': 964, 'HasAns_exact': 58.2537517053206, 'HasAns_f1': 78.24941928975495, 'HasAns_total': 733, 'NoAns_exact': 87.87878787878788, 'NoAns_f1': 87.87878787878788, 'NoAns_total': 231, 'best_exact': 65.35269709543569, 'best_exact_thresh': 0.0, 'best_f1': 80.55687172135933, 'best_f1_thresh': 0.0}
```

```
{'exact': 63.27800829875519, 'f1': 76.59012104331808, 'total': 964, 'HasAns_exact': 55.38881309686221, 'HasAns_f1': 72.89614827525055, 'HasAns_total': 733, 'NoAns_exact': 88.31168831168831, 'NoAns_f1': 88.31168831168831, 'NoAns_total': 231, 'best_exact': 63.27800829875519, 'best_exact_thresh': 0.0, 'best_f1': 76.5901210433181, 'best_f1_thresh': 0.0}
```

```
{'exact': 64.93775933609959, 'f1': 78.7738385522557, 'total': 964, 'HasAns_exact': 57.70804911323329, 'HasAns_f1': 75.90447526226122, 'HasAns_total': 733, 'NoAns_exact': 87.87878787878788, 'NoAns_f1': 87.87878787878788, 'NoAns_total': 231, 'best_exact': 64.93775933609959, 'best_exact_thresh': 0.0, 'best_f1': 78.7738385522557, 'best_f1_thresh': 0.0}
```

```
{'exact': 65.35269709543569, 'f1': 80.55687172135933, 'total': 964, 'HasAns_exact': 58.2537517053206, 'HasAns_f1': 78.24941928975495, 'HasAns_total': 733, 'NoAns_exact': 87.87878787878788, 'NoAns_f1': 87.87878787878788, 'NoAns_total': 231, 'best_exact': 65.35269709543569, 'best_exact_thresh': 0.0, 'best_f1': 80.55687172135933, 'best_f1_thresh': 0.0}
```

```
{'exact': 63.27800829875519, 'f1': 76.5968135878235, 'total': 964, 'HasAns_exact': 55.38881309686221, 'HasAns_f1': 72.90494992996163, 'HasAns_total': 733, 'NoAns_exact': 88.31168831168831, 'NoAns_f1': 88.31168831168831, 'NoAns_total': 231, 'best_exact': 63.27800829875519, 'best_exact_thresh': 0.0, 'best_f1': 76.59681358782353, 'best_f1_thresh': 0.0}
```

```
{'exact': 64.93775933609959, 'f1': 78.78053109973096, 'total': 964, 'HasAns_exact': 57.70804911323329, 'HasAns_f1': 75.9132769169723, 'HasAns_total': 733, 'NoAns_exact': 87.87878787878788, 'NoAns_f1': 87.87878787878788, 'NoAns_total': 231, 'best_exact': 64.93775933609959, 'best_exact_thresh': 0.0, 'best_f1': 78.780531099731, 'best_f1_thresh': 0.0}
```

```
{'exact': 65.35269709543569, 'f1': 80.55687172135933, 'total': 964, 'HasAns_exact': 58.2537517053206, 'HasAns_f1': 78.24941928975495, 'HasAns_total': 733, 'NoAns_exact': 87.87878787878788, 'NoAns_f1': 87.87878787878788, 'NoAns_total': 231, 'best_exact': 65.35269709543569, 'best_exact_thresh': 0.0, 'best_f1': 80.55687172135933, 'best_f1_thresh': 0.0}
```

خلاصه 9 حالت بالا برای دو خروجی 'exact' و 'f1' در جدول زیر آورده شده است :

	best_n	max_length	exact	f1
0	10.0	10.0	63.278008	76.707687
1	10.0	15.0	64.937759	78.773839
2	10.0	20.0	65.352697	80.556872
3	20.0	10.0	63.278008	76.590121
4	20.0	15.0	64.937759	78.773839

Fig15. Model's output comparison for prediction and reference

همچنین نتایج موجود در جدول بالا را در قالب دو نمودار مجزا برای exact و f1 در قالب scatter plot رسم کردیم :

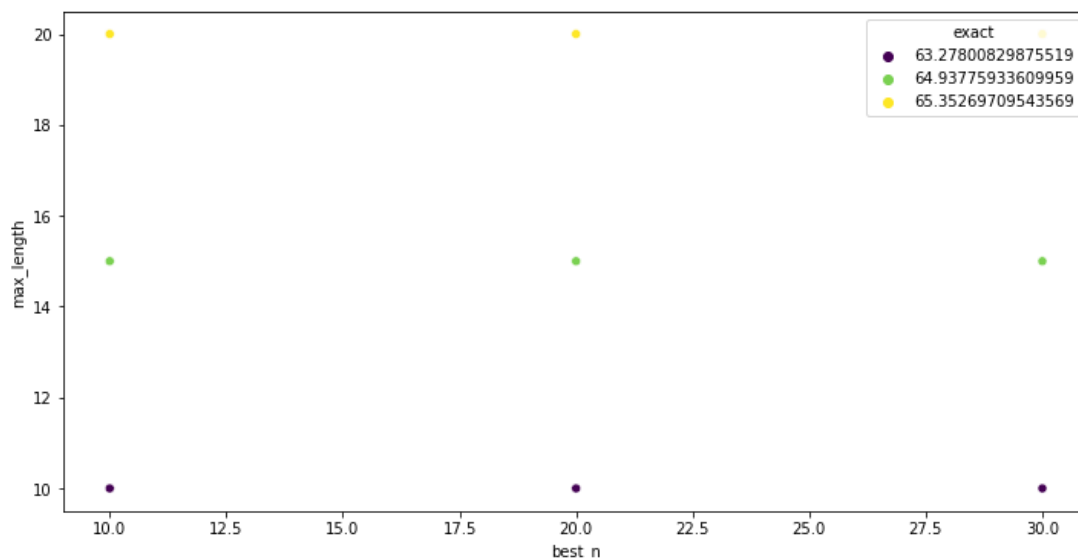


Fig16. exact metric performance for different ranges of best_n,max_length

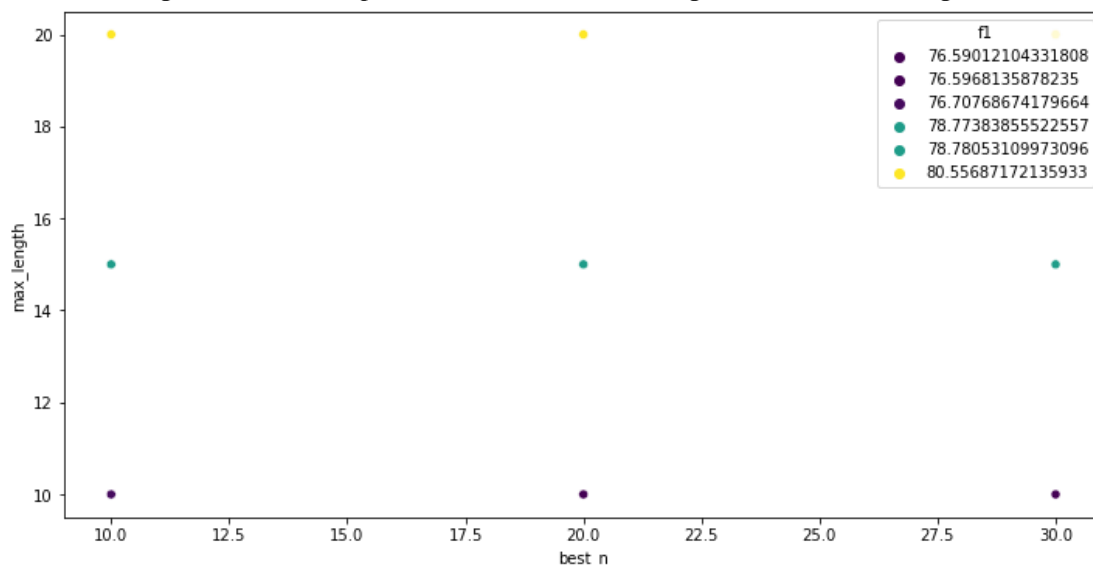


Fig17. exact metric performance for different ranges of best_n,max_length

همچنین پیشبینی مدل را برای یک id دلخواه با مقدار مرجع مقایسه می کنیم :

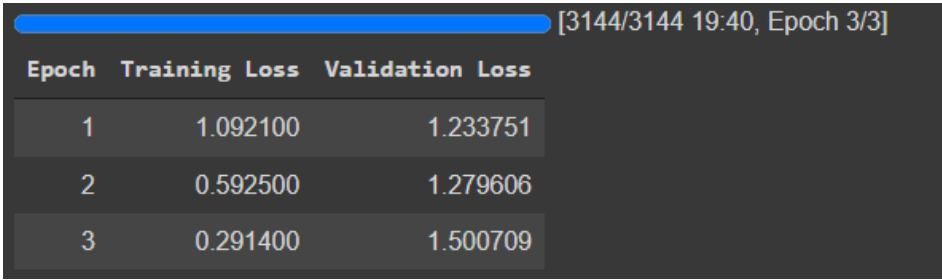
```
references[15]
{'answers': {'answer_start': [390], 'text': ['بااین موانع']}, 'id': 1719487.0}

formatted_predictions[15]
{'id': 1719487.0, 'no_answer_probability': 0.0, 'prediction_text': 'لینتراخت'}
```

Fig18. Model's output comparison for prediction and reference

PQuAD, PersianALBERT

بعد از epoch 3 آموزش مدل، نمودار خطای دادگان آموزش و ارزیابی به صورت زیر می‌باشد :



Epoch	Training Loss	Validation Loss
1	1.092100	1.233751
2	0.592500	1.279606
3	0.291400	1.500709

Fig19. Model loss for PersianALBERT on PQuAD

همچنین برای اینکه تاثیر دو پارامتر `n_best` و `max_length` را در عملکرد مدل بررسی کنیم، برای `n_best` از سه مقدار `[10,20,30]` و برای `max_length` از سه مقدار `[10,15,20]` استفاده می‌کنیم. پس در مجموع 9 حالت مختلف را بررسی می‌کنیم :

```
{'exact': 60.995850622406635, 'f1': 74.92763349809539, 'total': 964, 'HasAns_exact': 51.29604365620737,
'HasAns_f1': 69.61833382287043, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 61.09958506224066, 'best_exact_thresh': 0.0, 'best_f1':
75.03136793792942, 'best_f1_thresh': 0.0}
```

```
{'exact': 63.38174273858921, 'f1': 77.57309713028329, 'total': 964, 'HasAns_exact': 54.43383356070942,
'HasAns_f1': 73.09749745374235, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 63.48547717842324, 'best_exact_thresh': 0.0, 'best_f1':
77.67683157011732, 'best_f1_thresh': 0.0}
```

```
{'exact': 64.73029045643153, 'f1': 79.23612029666344, 'total': 964, 'HasAns_exact': 56.20736698499318,
'HasAns_f1': 75.28461114049601, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 64.83402489626556, 'best_exact_thresh': 0.0, 'best_f1':
79.33985473649747, 'best_f1_thresh': 0.0}
```

```
{'exact': 60.995850622406635, 'f1': 74.91810360889745, 'total': 964, 'HasAns_exact': 51.29604365620737,
'HasAns_f1': 69.60580065344776, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 61.09958506224066, 'best_exact_thresh': 0.0, 'best_f1':
75.02183804873148, 'best_f1_thresh': 0.0}
```

```
{'exact': 63.38174273858921, 'f1': 77.58431412905216, 'total': 964, 'HasAns_exact': 54.43383356070942,
'HasAns_f1': 73.11224941392403, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 63.48547717842324, 'best_exact_thresh': 0.0, 'best_f1':
77.68804856888617, 'best_f1_thresh': 0.0}
```

```
{'exact': 64.73029045643153, 'f1': 79.2610165622236, 'total': 964, 'HasAns_exact': 56.20736698499318,
'HasAns_f1': 75.31735329602125, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
```

91.77489177489177, 'NoAns_total': 231, 'best_exact': 64.83402489626556, 'best_exact_thresh': 0.0, 'best_f1': 79.36475100205764, 'best_f1_thresh': 0.0}

{'exact': 60.995850622406635, 'f1': 74.91810360889745, 'total': 964, 'HasAns_exact': 51.29604365620737, 'HasAns_f1': 69.60580065344776, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1': 91.77489177489177, 'NoAns_total': 231, 'best_exact': 61.09958506224066, 'best_exact_thresh': 0.0, 'best_f1': 75.02183804873148, 'best_f1_thresh': 0.0}

{'exact': 63.38174273858921, 'f1': 77.58431412905216, 'total': 964, 'HasAns_exact': 54.43383356070942, 'HasAns_f1': 73.11224941392403, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1': 91.77489177489177, 'NoAns_total': 231, 'best_exact': 63.48547717842324, 'best_exact_thresh': 0.0, 'best_f1': 77.68804856888617, 'best_f1_thresh': 0.0}

{'exact': 64.73029045643153, 'f1': 79.2610165622236, 'total': 964, 'HasAns_exact': 56.20736698499318, 'HasAns_f1': 75.31735329602125, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1': 91.77489177489177, 'NoAns_total': 231, 'best_exact': 64.83402489626556, 'best_exact_thresh': 0.0, 'best_f1': 79.36475100205764, 'best_f1_thresh': 0.0}

خلاصه 9 حالت بالا برای دو خروجی 'exact' و 'f1' در جدول زیر آورده شده است :

	best_n	max_length	exact	f1
0	10.0	10.0	60.995851	74.927633
1	10.0	15.0	63.381743	77.573097
2	10.0	20.0	64.730290	79.236120
3	20.0	10.0	60.995851	74.918104
4	20.0	15.0	63.381743	77.584314

Fig20. Model's output comparison for prediction and reference

همچنین نتایج موجود در جدول بالا را در قالب دو نمودار مجزا برای exact و f1 در قالب scatter plot رسم کردیم :

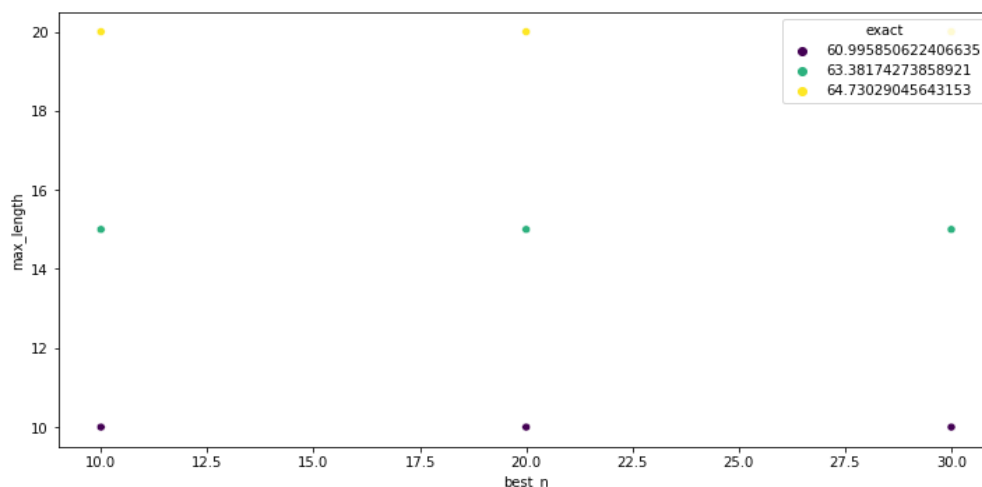


Fig21. exact metric performance for different ranges of best_n,max_length

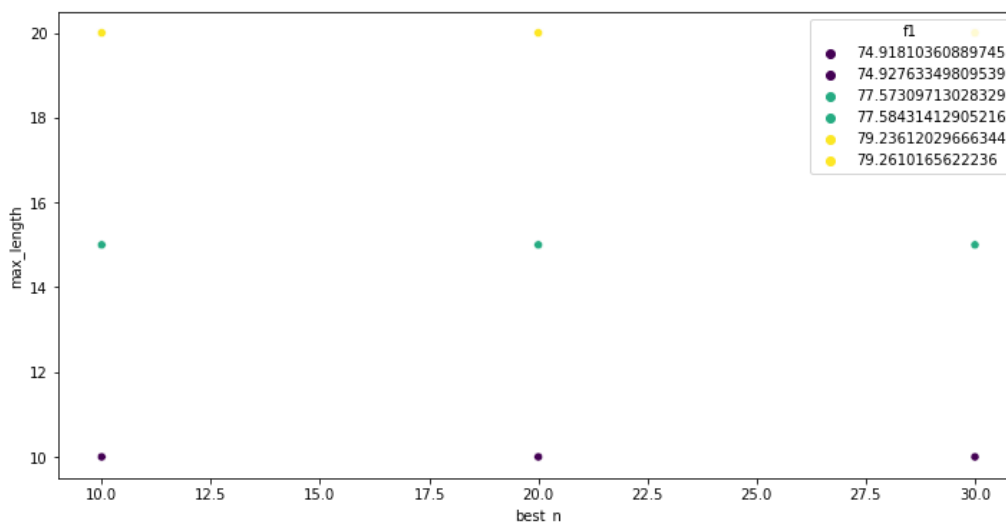


Fig21. f1 metric performance for different ranges of best_n,max_length

همچنین پیشبینی مدل را برای یک id دلخواه با مقدار مرجع مقایسه می‌کنیم :

```
references[15]
{'answers': {'answer_start': [390], 'text': ['بایرن مونیخ']}, 'id': 1719487.0}

formatted_predictions[15]
{'id': 1719487.0,
 'no_answer_probability': 0.0,
 'prediction_text': 'بایرن مونیخ'}
```

Fig22. Model's output comparison for prediction and reference

ParSQuAD,ParsBERT

بعد از epoch 3 آموزش مدل، نمودار خطای دادگان آموزش و ارزیابی به صورت زیر می‌باشد :

[7713/7713 54:31, Epoch 3/3]		
Epoch	Training Loss	Validation Loss
1	1.539900	1.704045
2	0.894400	1.845696
3	0.428000	2.691699

Fig23. Model loss for ParsBERT on ParSQuAD

همچنین برای اینکه تاثیر دو پارامتر n_best و max_length را در عملکرد مدل بررسی کنیم، برای n_best از سه مقدار [10,20,30] و برای max_length از سه مقدار [10,15,20] استفاده می‌کنیم. پس در مجموع 9 حالت مختلف را بررسی می‌کنیم :

{'exact': 51.6597510373444, 'f1': 63.96133908315053, 'total': 964, 'HasAns_exact': 48.43110504774898, 'HasAns_f1': 64.609455492711, 'HasAns_total': 733, 'NoAns_exact': 61.904761904761905, 'NoAns_f1': 61.904761904761905, 'NoAns_total': 231, 'best_exact': 51.76348547717842, 'best_exact_thresh': 0.0, 'best_f1': 64.06507352298456, 'best_f1_thresh': 0.0}

{'exact': 52.0746887966805, 'f1': 64.58790926462069, 'total': 964, 'HasAns_exact': 48.97680763983629, 'HasAns_f1': 65.43348503559946, 'HasAns_total': 733, 'NoAns_exact': 61.904761904761905, 'NoAns_f1': 61.904761904761905, 'NoAns_total': 231, 'best_exact': 52.178423236514526, 'best_exact_thresh': 0.0, 'best_f1': 64.69164370445473, 'best_f1_thresh': 0.0}

{'exact': 52.28215767634855, 'f1': 64.84156336704802, 'total': 964, 'HasAns_exact': 49.24965893587994, 'HasAns_f1': 65.76707651546296, 'HasAns_total': 733, 'NoAns_exact': 61.904761904761905, 'NoAns_f1': 61.904761904761905, 'NoAns_total': 231, 'best_exact': 52.385892116182575, 'best_exact_thresh': 0.0, 'best_f1': 64.94529780688204, 'best_f1_thresh': 0.0}

{'exact': 51.6597510373444, 'f1': 63.96133908315053, 'total': 964, 'HasAns_exact': 48.43110504774898, 'HasAns_f1': 64.609455492711, 'HasAns_total': 733, 'NoAns_exact': 61.904761904761905, 'NoAns_f1': 61.904761904761905, 'NoAns_total': 231, 'best_exact': 51.76348547717842, 'best_exact_thresh': 0.0, 'best_f1': 64.06507352298456, 'best_f1_thresh': 0.0}

{'exact': 52.0746887966805, 'f1': 64.58790926462069, 'total': 964, 'HasAns_exact': 48.97680763983629, 'HasAns_f1': 65.43348503559946, 'HasAns_total': 733, 'NoAns_exact': 61.904761904761905, 'NoAns_f1': 61.904761904761905, 'NoAns_total': 231, 'best_exact': 52.178423236514526, 'best_exact_thresh': 0.0, 'best_f1': 64.69164370445473, 'best_f1_thresh': 0.0}

{'exact': 52.28215767634855, 'f1': 64.84156336704802, 'total': 964, 'HasAns_exact': 49.24965893587994, 'HasAns_f1': 65.76707651546296, 'HasAns_total': 733, 'NoAns_exact': 61.904761904761905, 'NoAns_f1': 61.904761904761905, 'NoAns_total': 231, 'best_exact': 52.385892116182575, 'best_exact_thresh': 0.0, 'best_f1': 64.94529780688204, 'best_f1_thresh': 0.0}

{'exact': 51.6597510373444, 'f1': 63.96133908315053, 'total': 964, 'HasAns_exact': 48.43110504774898, 'HasAns_f1': 64.609455492711, 'HasAns_total': 733, 'NoAns_exact': 61.904761904761905, 'NoAns_f1': 61.904761904761905, 'NoAns_total': 231, 'best_exact': 51.76348547717842, 'best_exact_thresh': 0.0, 'best_f1': 64.06507352298456, 'best_f1_thresh': 0.0}

{'exact': 52.0746887966805, 'f1': 64.58790926462069, 'total': 964, 'HasAns_exact': 48.97680763983629, 'HasAns_f1': 65.43348503559946, 'HasAns_total': 733, 'NoAns_exact': 61.904761904761905, 'NoAns_f1': 61.904761904761905, 'NoAns_total': 231, 'best_exact': 52.178423236514526, 'best_exact_thresh': 0.0, 'best_f1': 64.69164370445473, 'best_f1_thresh': 0.0}

{'exact': 52.28215767634855, 'f1': 64.84156336704802, 'total': 964, 'HasAns_exact': 49.24965893587994, 'HasAns_f1': 65.76707651546296, 'HasAns_total': 733, 'NoAns_exact': 61.904761904761905, 'NoAns_f1':

61.904761904761905, 'NoAns_total': 231, 'best_exact': 52.385892116182575, 'best_exact_thresh': 0.0, 'best_f1': 64.94529780688204, 'best_f1_thresh': 0.0}

خلاصه 9 حالت بالا برای دو خروجی 'exact' و 'f1' در جدول زیر آورده شده است :

	best_n	max_length	exact	f1
0	10.0	10.0	51.659751	63.961339
1	10.0	15.0	52.074689	64.587909
2	10.0	20.0	52.282158	64.841563
3	20.0	10.0	51.659751	63.961339
4	20.0	15.0	52.074689	64.587909

Fig24. Model's output comparison for prediction and reference

همچنین نتایج موجود در جدول بالا را در قالب دو نمودار مجزا برای exact و f1 در قالب scatter plot رسم کردیم :

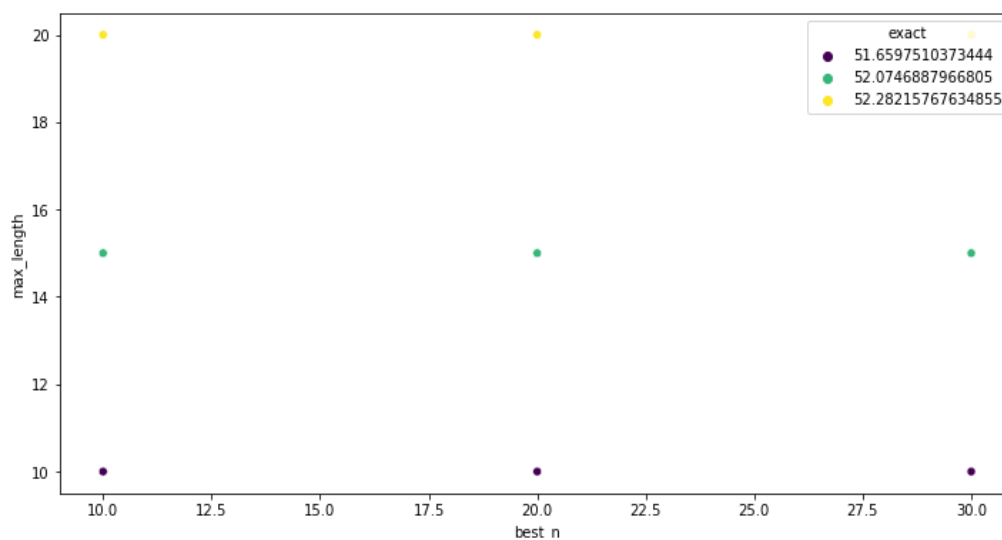


Fig25. exact metric performance for different ranges of best_n,max_length

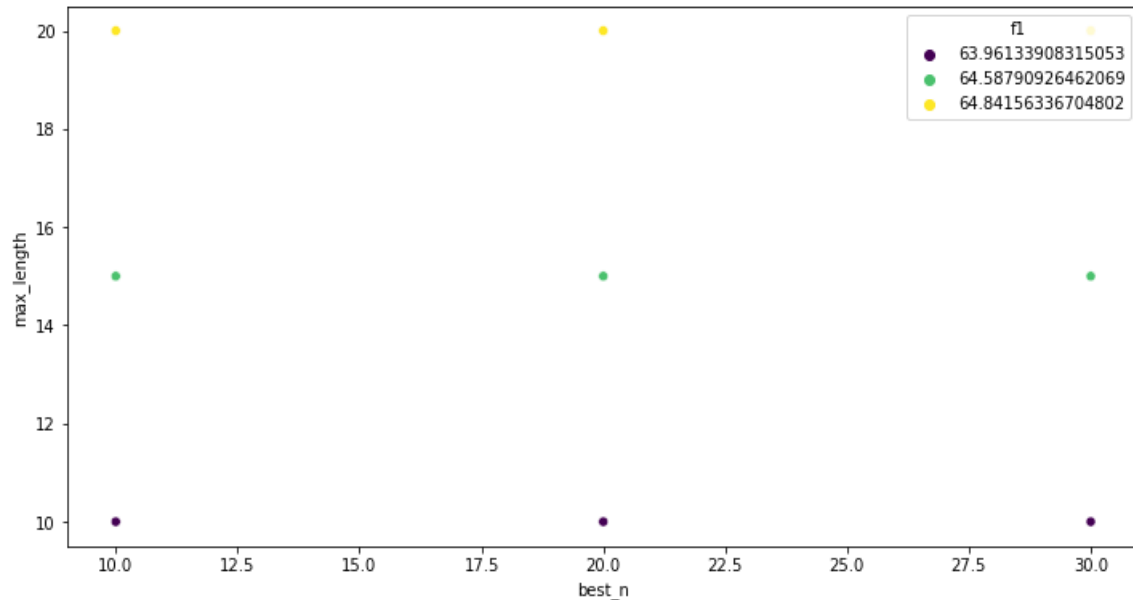


Fig26. f1 metric performance for different ranges of best_n,max_length

```
[ ] 1 references[15]
      {'answers': {'answer_start': [390], 'text': ['بايرن مونیخ']}, 'id': 1719487,0}

[▶] 1 formatted_predictions[15]
      {'id': 1719487,0, 'no_answer_probability': 0.0, 'prediction_text': 'بايرن'}
```

Fig27. Model's output comparison for prediction and reference

ParSQuAD,PersianALBERT

بعد از epoch 3 آموزش مدل، نمودار خطای دادگان آموزش و ارزیابی به صورت زیر می باشد :

[7938/7938 56:02, Epoch 3/3]		
Epoch	Training Loss	Validation Loss
1	1.434200	1.504050
2	0.752100	1.893190
3	0.332000	2.602487

Fig28. Model loss for PersianALBERT on ParSQuAD

همچنین برای اینکه تاثیر دو پارامتر n_best و max_length را در عملکرد مدل بررسی کنیم، برای n_best از سه مقدار [10,20,30] و برای max_length از سه مقدار [10,15,20] استفاده می کنیم. پس در مجموع 9 حالت مختلف را بررسی می کنیم :

{'exact': 49.79253112033195, 'f1': 62.21746803873967, 'total': 964, 'HasAns_exact': 43.24693042291951, 'HasAns_f1': 59.58750230470004, 'HasAns_total': 733, 'NoAns_exact': 70.56277056277057, 'NoAns_f1':

70.56277056277057, 'NoAns_total': 231, 'best_exact': 49.79253112033195, 'best_exact_thresh': 0.0, 'best_f1': 62.217468038739625, 'best_f1_thresh': 0.0}

{'exact': 50.4149377593361, 'f1': 63.23443678796301, 'total': 964, 'HasAns_exact': 44.065484311050476, 'HasAns_f1': 60.92496188758037, 'HasAns_total': 733, 'NoAns_exact': 70.56277056277057, 'NoAns_f1': 70.56277056277057, 'NoAns_total': 231, 'best_exact': 50.4149377593361, 'best_exact_thresh': 0.0, 'best_f1': 63.23443678796297, 'best_f1_thresh': 0.0}

{'exact': 50.31120331950208, 'f1': 63.30674639103668, 'total': 964, 'HasAns_exact': 43.92905866302865, 'HasAns_f1': 61.02005937375091, 'HasAns_total': 733, 'NoAns_exact': 70.56277056277057, 'NoAns_f1': 70.56277056277057, 'NoAns_total': 231, 'best_exact': 50.31120331950208, 'best_exact_thresh': 0.0, 'best_f1': 63.30674639103663, 'best_f1_thresh': 0.0}

{'exact': 49.79253112033195, 'f1': 62.21746803873967, 'total': 964, 'HasAns_exact': 43.24693042291951, 'HasAns_f1': 59.58750230470004, 'HasAns_total': 733, 'NoAns_exact': 70.56277056277057, 'NoAns_f1': 70.56277056277057, 'NoAns_total': 231, 'best_exact': 49.79253112033195, 'best_exact_thresh': 0.0, 'best_f1': 62.217468038739625, 'best_f1_thresh': 0.0}

{'exact': 50.4149377593361, 'f1': 63.23443678796301, 'total': 964, 'HasAns_exact': 44.065484311050476, 'HasAns_f1': 60.92496188758037, 'HasAns_total': 733, 'NoAns_exact': 70.56277056277057, 'NoAns_f1': 70.56277056277057, 'NoAns_total': 231, 'best_exact': 50.4149377593361, 'best_exact_thresh': 0.0, 'best_f1': 63.23443678796297, 'best_f1_thresh': 0.0}

{'exact': 50.31120331950208, 'f1': 63.30674639103668, 'total': 964, 'HasAns_exact': 43.92905866302865, 'HasAns_f1': 61.02005937375091, 'HasAns_total': 733, 'NoAns_exact': 70.56277056277057, 'NoAns_f1': 70.56277056277057, 'NoAns_total': 231, 'best_exact': 50.31120331950208, 'best_exact_thresh': 0.0, 'best_f1': 63.30674639103663, 'best_f1_thresh': 0.0}

{'exact': 49.79253112033195, 'f1': 62.21746803873967, 'total': 964, 'HasAns_exact': 43.24693042291951, 'HasAns_f1': 59.58750230470004, 'HasAns_total': 733, 'NoAns_exact': 70.56277056277057, 'NoAns_f1': 70.56277056277057, 'NoAns_total': 231, 'best_exact': 49.79253112033195, 'best_exact_thresh': 0.0, 'best_f1': 62.217468038739625, 'best_f1_thresh': 0.0}

{'exact': 50.4149377593361, 'f1': 63.23443678796301, 'total': 964, 'HasAns_exact': 44.065484311050476, 'HasAns_f1': 60.92496188758037, 'HasAns_total': 733, 'NoAns_exact': 70.56277056277057, 'NoAns_f1': 70.56277056277057, 'NoAns_total': 231, 'best_exact': 50.4149377593361, 'best_exact_thresh': 0.0, 'best_f1': 63.23443678796297, 'best_f1_thresh': 0.0}

{'exact': 50.31120331950208, 'f1': 63.30674639103668, 'total': 964, 'HasAns_exact': 43.92905866302865, 'HasAns_f1': 61.02005937375091, 'HasAns_total': 733, 'NoAns_exact': 70.56277056277057, 'NoAns_f1': 70.56277056277057, 'NoAns_total': 231, 'best_exact': 50.31120331950208, 'best_exact_thresh': 0.0, 'best_f1': 63.30674639103663, 'best_f1_thresh': 0.0}

خلاصه 9 حالت بالا برای دو خروجی 'exact' و 'f1' در جدول زیر آورده شده است :

	best_n	max_length	exact	f1
0	10.0	10.0	49.792531	62.217468
1	10.0	15.0	50.414938	63.234437
2	10.0	20.0	50.311203	63.306746
3	20.0	10.0	49.792531	62.217468
4	20.0	15.0	50.414938	63.234437

Fig29. Model's output comparison for prediction and reference

همچنین نتایج موجود در جدول بالا را در قالب دو نمودار مجزا برای exact و f1 در قالب scatter plot رسم کردیم :

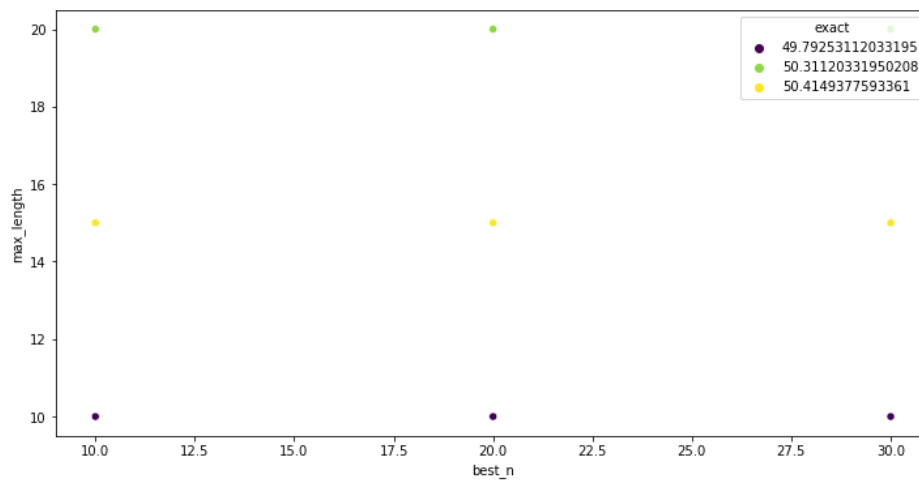


Fig30. exact metric performance for different ranges of best_n,max_length

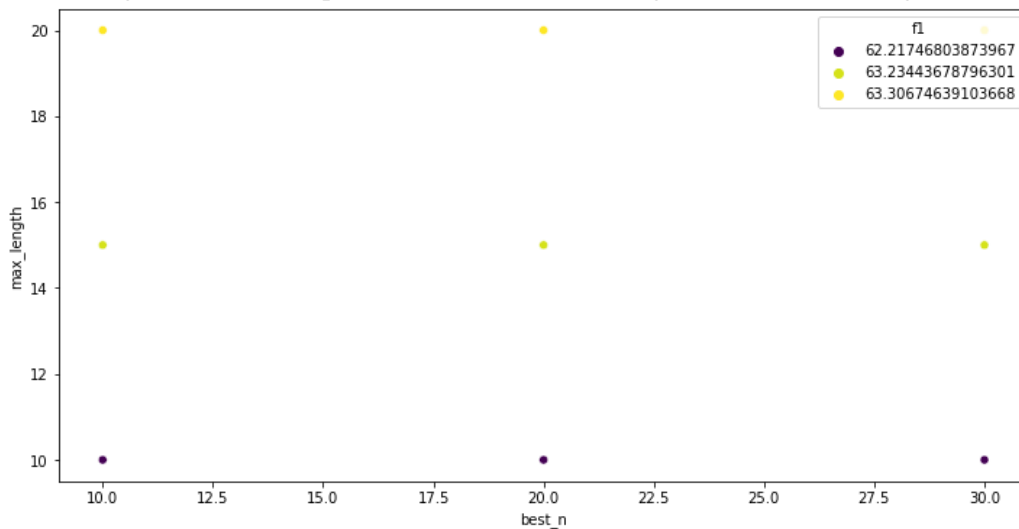


Fig31. exact metric performance for different ranges of best_n,max_length

همچنین پیشبینی مدل را برای یک id دلخواه با مقدار مرجع مقایسه می کنیم :

```

references[15]

{'answers': {'answer_start': [390], 'text': ['بایرن مونیخ']}, 'id': 1719487.0}

formatted_predictions[15]

{'id': 1719487.0,
 'no_answer_probability': 0.0,
 'prediction_text': 'بایرن مونیخ'}

```

Fig32. Model's output comparison for prediction and reference

(PQuAD + PersianQA), ParsBERT

بعد از epoch 3 آموزش مدل، نمودار خطای دادگان آموزش و ارزیابی به صورت زیر می باشد :

[6873/6873 47:58, Epoch 3/3]

Epoch	Training Loss	Validation Loss
1	1.527100	1.318570
2	0.835100	1.353809
3	0.401400	1.746275

Fig33. Model loss for ParsBERT on (PQuAD + PersianQA)

همچنین برای اینکه تاثیر دو پارامتر `n_best` و `max_length` را در عملکرد مدل بررسی کنیم، برای `n_best` از سه مقدار `[10,20,30]` و برای `max_length` از سه مقدار `[10,15,20]` استفاده می کنیم. پس در مجموع 9 حالت مختلف را بررسی می کنیم :

```
{'exact': 64.83402489626556, 'f1': 78.77704672991916, 'total': 964, 'HasAns_exact': 56.343792633015006,
'HasAns_f1': 74.680863639348, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 64.83402489626556, 'best_exact_thresh': 0.0, 'best_f1':
78.77704672991918, 'best_f1_thresh': 0.0}
```

```
{'exact': 67.32365145228216, 'f1': 81.64029876131966, 'total': 964, 'HasAns_exact': 59.61800818553888,
'HasAns_f1': 78.44645021270415, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 67.32365145228216, 'best_exact_thresh': 0.0, 'best_f1':
81.64029876131966, 'best_f1_thresh': 0.0}
```

```
{'exact': 67.73858921161826, 'f1': 82.801897503487, 'total': 964, 'HasAns_exact': 60.16371077762619,
'HasAns_f1': 79.9741189541084, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 67.73858921161826, 'best_exact_thresh': 0.0, 'best_f1':
82.80189750348698, 'best_f1_thresh': 0.0}
```

```
{'exact': 64.83402489626556, 'f1': 78.79446408356903, 'total': 964, 'HasAns_exact': 56.343792633015006,
'HasAns_f1': 74.70376995438004, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 64.83402489626556, 'best_exact_thresh': 0.0, 'best_f1':
78.79446408356905, 'best_f1_thresh': 0.0}
```

```
{'exact': 67.32365145228216, 'f1': 81.64029876131966, 'total': 964, 'HasAns_exact': 59.61800818553888,
'HasAns_f1': 78.44645021270415, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 67.32365145228216, 'best_exact_thresh': 0.0, 'best_f1':
81.64029876131966, 'best_f1_thresh': 0.0}
```

```
{'exact': 67.73858921161826, 'f1': 82.801897503487, 'total': 964, 'HasAns_exact': 60.16371077762619,
'HasAns_f1': 79.9741189541084, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 67.73858921161826, 'best_exact_thresh': 0.0, 'best_f1':
82.80189750348698, 'best_f1_thresh': 0.0}
```

```
{'exact': 64.83402489626556, 'f1': 78.79446408356903, 'total': 964, 'HasAns_exact': 56.343792633015006,
'HasAns_f1': 74.70376995438004, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 64.83402489626556, 'best_exact_thresh': 0.0, 'best_f1':
78.79446408356905, 'best_f1_thresh': 0.0}
```

```
{'exact': 67.32365145228216, 'f1': 81.64029876131966, 'total': 964, 'HasAns_exact': 59.61800818553888,
'HasAns_f1': 78.44645021270415, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 67.32365145228216, 'best_exact_thresh': 0.0, 'best_f1':
81.64029876131966, 'best_f1_thresh': 0.0}
```

```
{'exact': 67.73858921161826, 'f1': 82.801897503487, 'total': 964, 'HasAns_exact': 60.16371077762619,
'HasAns_f1': 79.9741189541084, 'HasAns_total': 733, 'NoAns_exact': 91.77489177489177, 'NoAns_f1':
91.77489177489177, 'NoAns_total': 231, 'best_exact': 67.73858921161826, 'best_exact_thresh': 0.0, 'best_f1':
82.80189750348698, 'best_f1_thresh': 0.0}
```

خلاصه 9 حالت بالا برای دو خروجی 'exact' و 'f1' در جدول زیر آورده شده است :

	best_n	max_length	exact	f1
0	10.0	10.0	64.834025	78.777047
1	10.0	15.0	67.323651	81.640299
2	10.0	20.0	67.738589	82.801898
3	20.0	10.0	64.834025	78.794464
4	20.0	15.0	67.323651	81.640299

Fig34. Model's output comparison for prediction and reference

همچنین نتایج موجود در جدول بالا را در قالب دو نمودار مجزا برای exact و f1 در قالب scatter plot رسم کردیم :

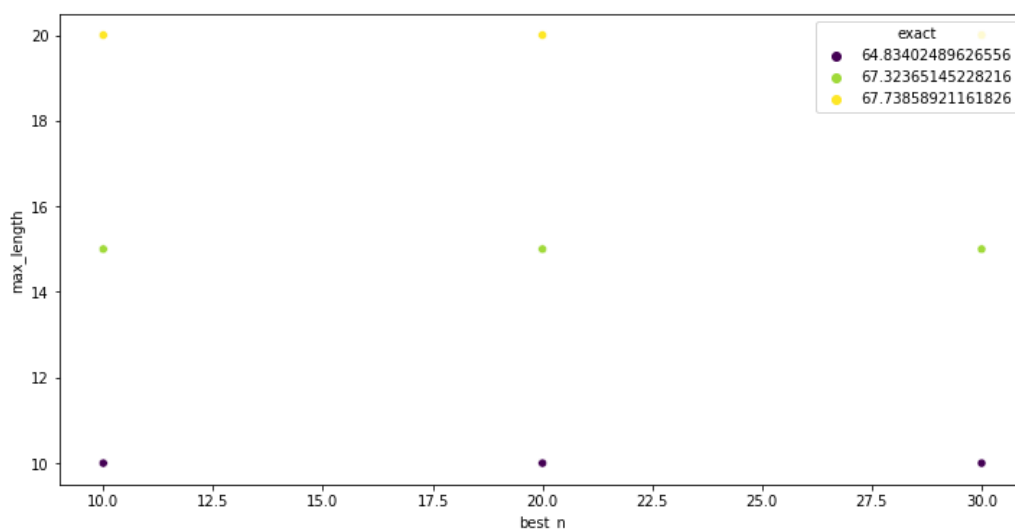


Fig35. exact metric performance for different ranges of best_n,max_length

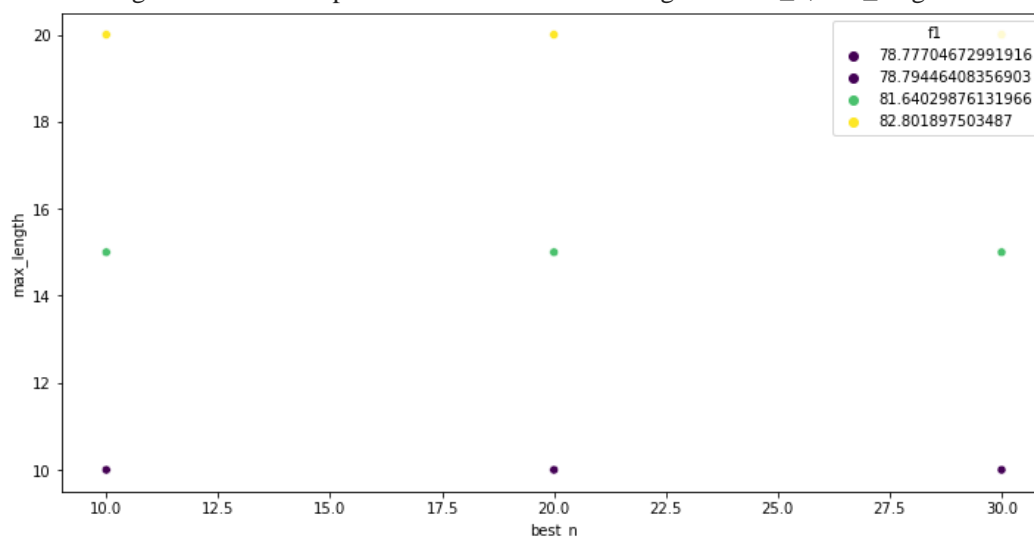


Fig36. f1 metric performance for different ranges of best_n,max_length

همچنین پیشبینی مدل را برای یک id دلخواه با مقدار مرجع مقایسه می‌کنیم :

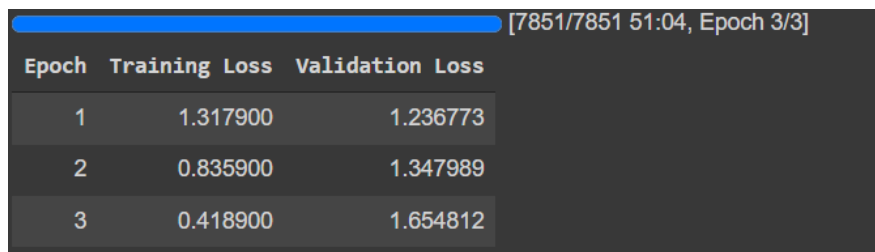
```
references[15]
{'answers': {'answer_start': [390], 'text': ['بایرن مونیخ']}, 'id': 1719487.0}

formatted_predictions[15]
{'id': 1719487.0,
 'no_answer_probability': 0.0,
 'prediction_text': 'بایرن مونیخ'}
```

Fig37. Model's output comparison for prediction and reference

(PQuAD + PersianQA), ALBERTPersian

بعد از epoch 3 آموزش مدل، نمودار خطای دادگان آموزش و ارزیابی به صورت زیر می‌باشد:



Epoch	Training Loss	Validation Loss
1	1.317900	1.236773
2	0.835900	1.347989
3	0.418900	1.654812

Fig38. Model loss for ALBERTPersian on (PQuAD + PersianQA)

همچنین برای اینکه تاثیر دو پارامتر `n_best` و `max_length` را در عملکرد مدل بررسی کنیم، برای `n_best` از سه مقدار `[10,20,30]` و برای `max_length` از سه مقدار `[10,15,20]` استفاده می‌کنیم. پس در مجموع 9 حالت مختلف را بررسی می‌کنیم:

```
{'exact': 62.344398340248965, 'f1': 76.31715919670188, 'total': 964, 'HasAns_exact': 52.796725784447474, 'HasAns_f1': 71.1729078657853, 'HasAns_total': 733, 'NoAns_exact': 92.64069264069263, 'NoAns_f1': 92.64069264069263, 'NoAns_total': 231, 'best_exact': 62.344398340248965, 'best_exact_thresh': 0.0, 'best_f1': 76.31715919670187, 'best_f1_thresh': 0.0}
```

```
{'exact': 64.6265560165975, 'f1': 78.95817582150268, 'total': 964, 'HasAns_exact': 55.7980900409277, 'HasAns_f1': 74.64622304492302, 'HasAns_total': 733, 'NoAns_exact': 92.64069264069263, 'NoAns_f1': 92.64069264069263, 'NoAns_total': 231, 'best_exact': 64.6265560165975, 'best_exact_thresh': 0.0, 'best_f1': 78.95817582150265, 'best_f1_thresh': 0.0}
```

```
{'exact': 65.56016597510373, 'f1': 80.38241843665504, 'total': 964, 'HasAns_exact': 57.16234652114598, 'HasAns_f1': 76.65573175025305, 'HasAns_total': 733, 'NoAns_exact': 92.20779220779221, 'NoAns_f1': 92.20779220779221, 'NoAns_total': 231, 'best_exact': 65.56016597510373, 'best_exact_thresh': 0.0, 'best_f1': 80.38241843665499, 'best_f1_thresh': 0.0}
```

```
{'exact': 62.344398340248965, 'f1': 76.2921818289585, 'total': 964, 'HasAns_exact': 52.796725784447474, 'HasAns_f1': 71.14005904927151, 'HasAns_total': 733, 'NoAns_exact': 92.64069264069263, 'NoAns_f1': 92.64069264069263, 'NoAns_total': 231, 'best_exact': 62.344398340248965, 'best_exact_thresh': 0.0, 'best_f1': 76.2921818289585, 'best_f1_thresh': 0.0}
```

```
{'exact': 64.6265560165975, 'f1': 78.91308603515806, 'total': 964, 'HasAns_exact': 55.7980900409277, 'HasAns_f1': 74.58692351690637, 'HasAns_total': 733, 'NoAns_exact': 92.64069264069263, 'NoAns_f1': 92.64069264069263, 'NoAns_total': 231, 'best_exact': 64.6265560165975, 'best_exact_thresh': 0.0, 'best_f1': 78.91308603515802, 'best_f1_thresh': 0.0}
```

```
{'exact': 65.56016597510373, 'f1': 80.38241843665504, 'total': 964, 'HasAns_exact': 57.16234652114598,
'HasAns_f1': 76.65573175025305, 'HasAns_total': 733, 'NoAns_exact': 92.20779220779221, 'NoAns_f1':
92.20779220779221, 'NoAns_total': 231, 'best_exact': 65.56016597510373, 'best_exact_thresh': 0.0, 'best_f1':
80.38241843665499, 'best_f1_thresh': 0.0}
```

```
{'exact': 62.344398340248965, 'f1': 76.2921818289585, 'total': 964, 'HasAns_exact': 52.796725784447474,
'HasAns_f1': 71.14005904927151, 'HasAns_total': 733, 'NoAns_exact': 92.64069264069263, 'NoAns_f1':
92.64069264069263, 'NoAns_total': 231, 'best_exact': 62.344398340248965, 'best_exact_thresh': 0.0, 'best_f1':
76.2921818289585, 'best_f1_thresh': 0.0}
```

```
{'exact': 64.6265560165975, 'f1': 78.91308603515806, 'total': 964, 'HasAns_exact': 55.7980900409277,
'HasAns_f1': 74.58692351690637, 'HasAns_total': 733, 'NoAns_exact': 92.64069264069263, 'NoAns_f1':
92.64069264069263, 'NoAns_total': 231, 'best_exact': 64.6265560165975, 'best_exact_thresh': 0.0, 'best_f1':
78.91308603515802, 'best_f1_thresh': 0.0}
```

```
{'exact': 65.56016597510373, 'f1': 80.38241843665504, 'total': 964, 'HasAns_exact': 57.16234652114598,
'HasAns_f1': 76.65573175025305, 'HasAns_total': 733, 'NoAns_exact': 92.20779220779221, 'NoAns_f1':
92.20779220779221, 'NoAns_total': 231, 'best_exact': 65.56016597510373, 'best_exact_thresh': 0.0, 'best_f1':
80.38241843665499, 'best_f1_thresh': 0.0}
```

همچنین نتایج موجود در جدول بالا را در قالب دو نمودار مجزا برای exact و f1 در قالب scatter plot رسم کردیم :

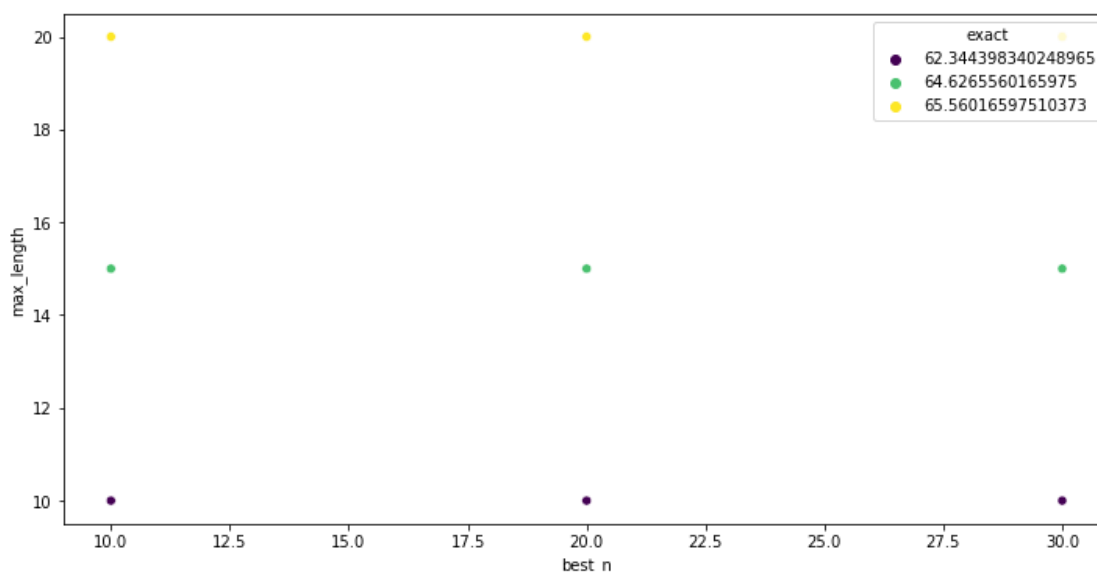


Fig39. exact metric performance for different ranges of best_n,max_length

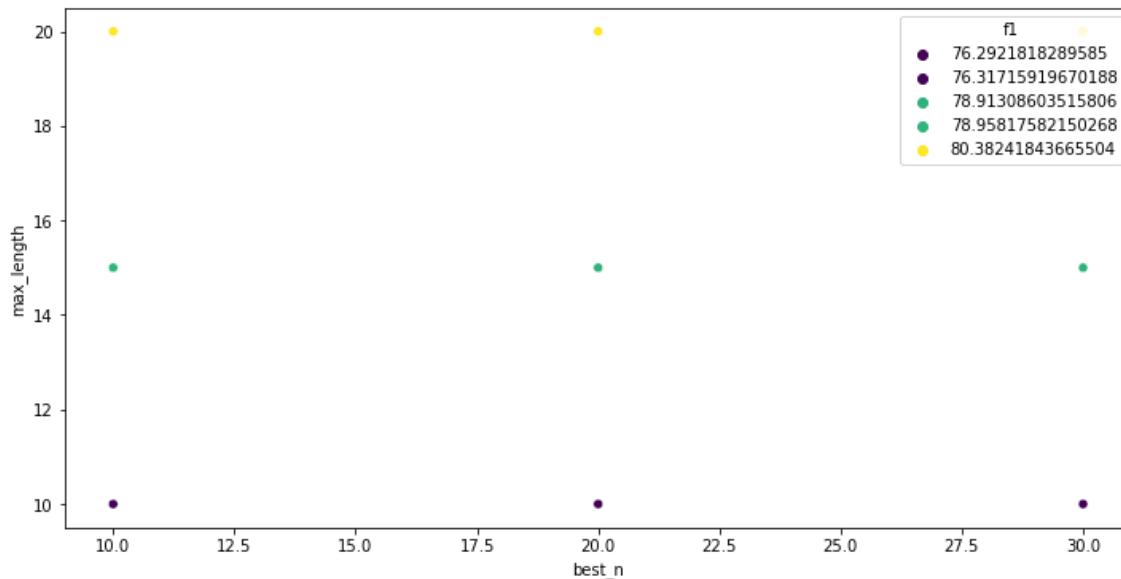


Fig40. f1 metric performance for different ranges of best_n,max_length

همچنین پیشبینی مدل را برای یک id دلخواه با مقدار مرجع مقایسه می‌کنیم :

```
references[15]

{'answers': {'answer_start': [390], 'text': ['بایرن مونیخ']}, 'id': 1719487.0}

formatted_predictions[15]

{'id': 1719487.0,
 'no_answer_probability': 0.0,
 'prediction_text': 'بایرن مونیخ'}
```

Fig41. Model's output comparison for prediction and reference

Conclusion

1. طبق نتایج بدست آمده در حین آموزش مدل، خطای دادگان آموزش رو به کاهش است ولی خطای دادگان ارزیابی رو به افزایش است بنابراین مدل در حال overfit شدن می‌باشد.
2. بهترین نتیجه بدست آمده مربوط به ParsBERT (PQuAD + PersianQA)، که بهترین مقدار f1 برابر 82 و بهترین مقدار exact برابر 67.74 می‌باشد.
3. بدترین نتیجه بدست آمده مربوط به ParSQuAD, PersianALBERT که بهترین f1 برابر 63.3 و بهترین exact برابر 50.41 می‌باشد.
4. تقریباً در همه حالت با افزایش max_length نتایج بهتر می‌شود و با تغییر n_best نتایج خیلی تغییر محسوسی نمی‌کند این نشان می‌دهد برای n_best بزرگتر از 10 تقریباً تغییری در عملکرد مدل ایجاد نمی‌شود زیرا احتمال اینکه

خروجی بهتری با انتخاب Token هایی با rank بیشتر از 10 پیدا شود بسیار ناچیز است ولی پارامتر max_length نسبت مستقیمی در عملکرد مدل دارد.

5. در کل نتایج هنگامی که از دیتاست ParSQuAD استفاده می‌کنیم افت شدیدی می‌کند زیرا اولاً از حالت manual استفاده کردیم که تعداد داده‌های آن بسیار کمتر از حالت automatic است و همچنین نوع لیبل زنی جملات در این حالت خیلی دقیق نیست و همچنین متن موجود در هر سطر غلط‌های زیادی از نظر معنایی دارد.
6. در آخر اگر از مدل‌های چند زبانه مثل XLM-RoBERTa که در بخش دوم نیز استفاده شد، بهره بگیریم و بر روی دیتاست اصلی SQuAD هم کار fine tuning انجام دهیم، می‌توانیم از دانشی که مدل از آن کسب می‌کند استفاده کرده و امیدوار باشیم که نتایج حاصل نسبت به مدل‌های PersianALBERT و ParsBERT بهتر شود.

References

- [1] Kasra Darvishi, Newsha Shahbodagh, Zahra Abbasiantaeb, Saeedeh Momtazi: “PQuAD: A Persian Question Answering Dataset”, 2022; arXiv:2202.06219.
- [2] N. Abadani, J. Mozafari, A. Fatemi, M. A. Nematbakhsh, and A. Kazemi, ‘ParSQuAD: Machine Translated SQuAD dataset for Persian Question Answering’, in 2021 7th International Conference on Web Research (ICWR), 2021, pp. 163–168. doi: 10.1109/ICWR51868.2021.9443126.
- [3] N. Abadani, J. Mozafari, A. Fatemi, M. Nematbakhsh, and A. Kazemi, ‘ParSQuAD: Persian Question Answering Dataset based on Machine Translation of SQuAD 2.0’, International Journal of Web Research, vol. 4, no. 1, Art. no. 1, 2021, doi: 10.22133/IJWR.2021.293313.1101.
- [4] S. Ayoubi, M. Y. Davoodeh, ‘PersianQA: a dataset for Persian Question Answering’, GitHub repository, 2021.
- [5] Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, Mohammad Manthouri: “ParsBERT: Transformer-based Model for Persian Language Understanding”, 2020; arXiv:2005.12515. DOI: 10.1007/s11063-021-10528-4.
- [6] Hooshvare Team, ‘ALBERT-Persian: A Lite BERT for Self-supervised Learning of Language Representations for the Persian Language’, GitHub repository, 2021.

PART 2 - Natural Language Understanding

Abstract

در این قسمت به یکی از تسک‌های NLP که امروزه با پیشرفت چت‌بات‌ها، دستیارهای صوتی و به صورت کلی دیالوگ سیستم‌ها، به یکی از موضوعات داغ تبدیل شده است می‌پردازیم. در تسک NLU دو عملیات مهم یعنی تشخیص قصد و نیت (intent) کاربر از بیان یک جمله و نیز تشخیص موجودیت‌های اسمی (Named entity) موجود در جمله کاربر باید انجام شود. مثلاً intent جمله مثال زیر که موجودیت‌ها و slot‌های آن نیز مشخص شده‌اند، تنظیم رویداد در تقویم (calendar_set) است؛ یک چت‌بات یا دستیار صوتی لازم است تا این اطلاعات را از جمله کاربر بدست آورد تا کاری که کاربر از او خواسته را به درستی متوجه شده و اجرا کند و اشتباه در تشخیص هر کدام از این موارد باعث نارضایتی کاربر بوده و ممکن است ضررهای احتمالی (نظیر از دست دادن ملاقات به دلیل عدم تنظیم درست تقویم یا ...) نیز برای کاربر ایجاد کند. بنابراین تسک NLU یک موضوع بسیار حیاتی و در عین حال پیچیده است که حتی انسان‌ها هم ممکن است در انجام آن دچار خطا شوند چه رسد به یک مدل شبکه عصبی مصنوعی. اما امروزه با پیشرفت مدل‌های شبکه عصبی عمیق و معرفی معماری‌های نظیر ترنسفورمرها، پیچیده‌ترین تسک‌ها نیز توسط این مدل‌ها با دقت خوبی قابل انجام است.

مثال از یک utterance کاربر و اطلاعاتی که یک دیالوگ سیستم باید از آن استخراج کند:

یک یادآور برای [event_name: چاپ اسناد] در [timeofday: صبح : date] [سه شنبه] تنظیم کن.

intent: calendar_set

Dataset

در اینجا ما از دیتاست MASSIVE که اخیراً توسط شرکت آمازون ارائه شده است استفاده می‌کنیم. این دیتاست شامل بیش از یک میلیون جمله در 51 زبان مختلف از جمله زبان فارسی است که برای تسک NLU تهیه شده‌اند. جملات این دیتاست از 18 domain مختلف هستند که شامل 60 intent در زمینه‌های مختلف نظیر 'weather_query', 'iot_hue_lightdim', 'calendar_set' و ... است. همچنین هر جمله می‌تواند 55 slot مختلف نظیر 'event_name', 'meal_type', 'timeofday', 'business_name' را شامل شود. در این پروژه ما صرفاً از دیتاست مربوط به زبان فارسی آن استفاده کردیم که به ترتیب 11514، 2033، 2974 نمونه به عنوان داده تست، ارزیابی و آموزش دارد.

id	locale	partition	scenario	intent	utt	annot_utt	worker_id	slot_method	judgments
1	1	fa-IR	train	alarm	alarm_set	مرا جمعه ساعت نه صبح بیدار کن	3	[[{'slot': 'time', 'method': 'translation'}, {'slot': 'date', 'method': 'translation'}], {'slot': 'intent', 'method': 'translation'}]	[[{'worker_id': '3', 'intent_score': 1, 'slots_score': 1}]]
2	2	fa-IR	train	alarm	alarm_set	یک زنگ هشدار را برای دو ساعت دیگر تنظیم کن	8	[[{'slot': 'time', 'method': 'translation'}], {'slot': 'intent', 'method': 'translation'}]	[[{'worker_id': '21', 'intent_score': 1, 'slots_score': 1}]]
4	4	fa-IR	train	audio	audio_volume_mute	آلی ساکت شو	21	[[{'slot': 'intent', 'method': 'translation'}]]	[[{'worker_id': '21', 'intent_score': 1, 'slots_score': 1}]]
5	5	fa-IR	train	audio	audio_volume_mute	توقف	17	[[{'slot': 'intent', 'method': 'translation'}]]	[[{'worker_id': '3', 'intent_score': 1, 'slots_score': 1}]]
6	6	fa-IR	train	audio	audio_volume_mute	برای ده ثانیه متوقف کن	17	[[{'slot': 'time', 'method': 'translation'}], {'slot': 'intent', 'method': 'translation'}]	[[{'worker_id': '8', 'intent_score': 1, 'slots_score': 1}]]

Fig1. A few samples from MASSIVE dataset (locale is fa-IR only)

توزیع نمونه‌های فارسی این دیتاست در کلاس‌های مختلف intent و نیز کلاس‌های مختلف slot در نمودارهای ذیل آمده است. همچنین مقادیر عددی و نسبی (درصدی) متناظر با هر کدام از نمودارها در بخش پیوست در پایان گزارش آمده است.

- توزیع intent نمونه‌های کل دیتاست (آموزش، ارزیابی و تست روی هم):

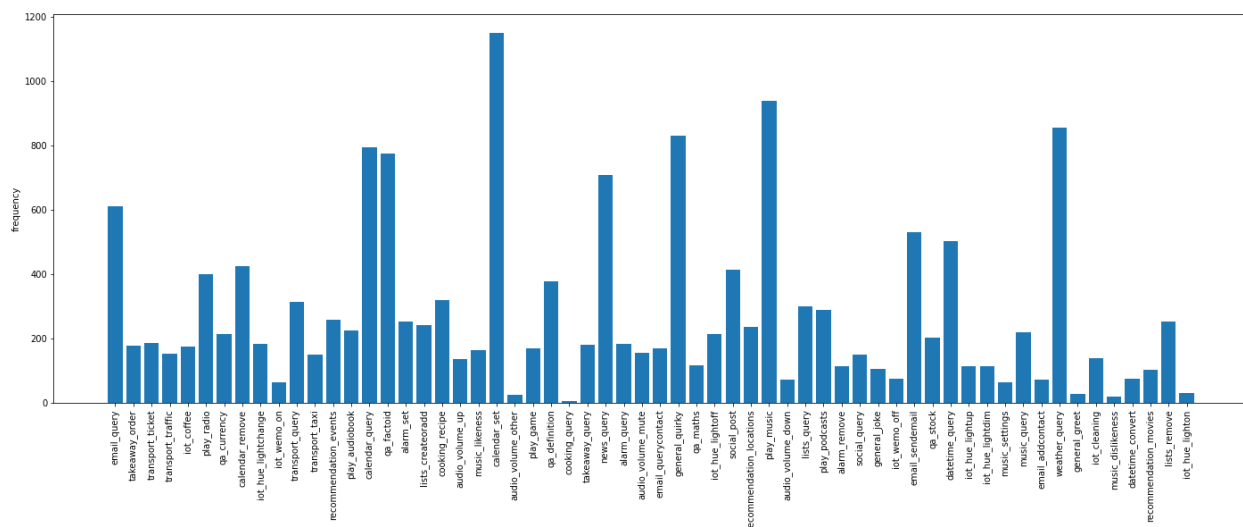


Fig2. Intents distribution in the whole dataset

- توزیع intent نمونه‌های دادگان آموزش:

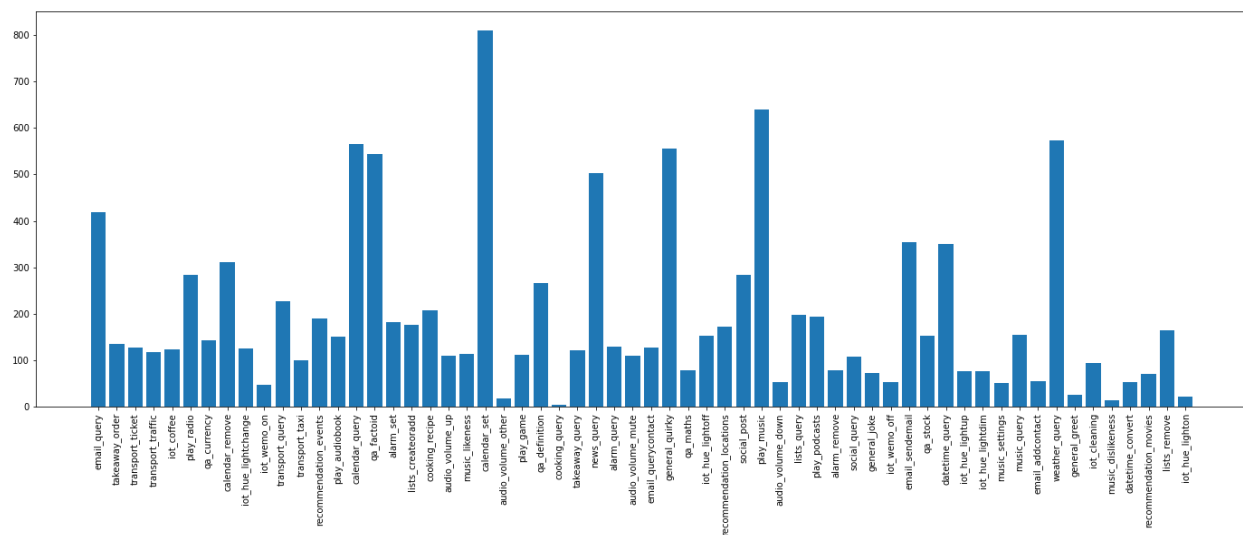


Fig3. Intents distribution in the train dataset

- توزیع intent نمونه‌های دادگان ارزیابی:

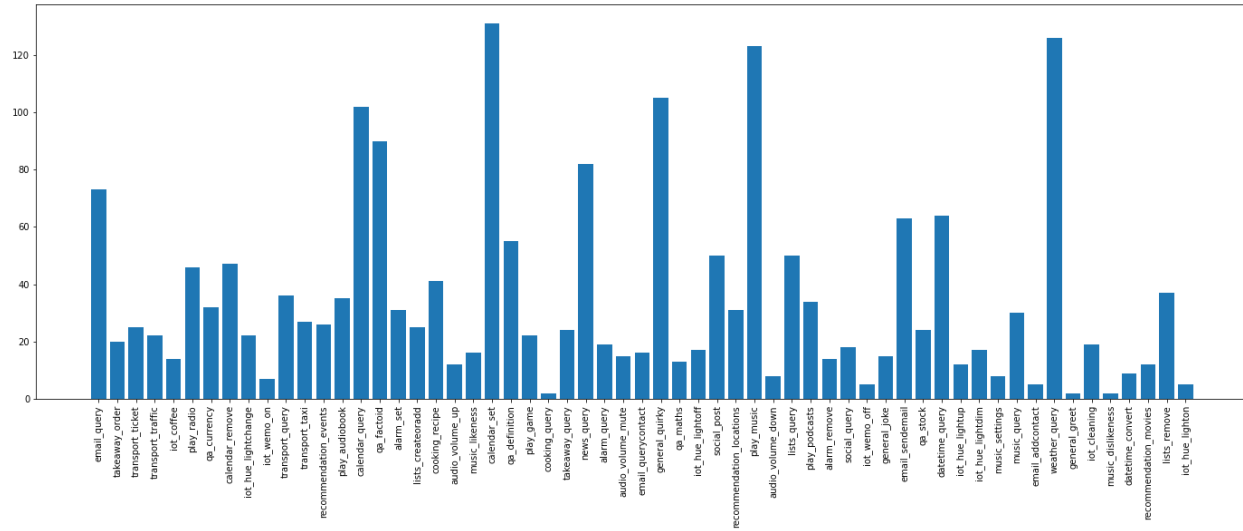


Fig4. Intents distribution in the eval dataset

● توزیع intent نمونه‌های دادگان تست:

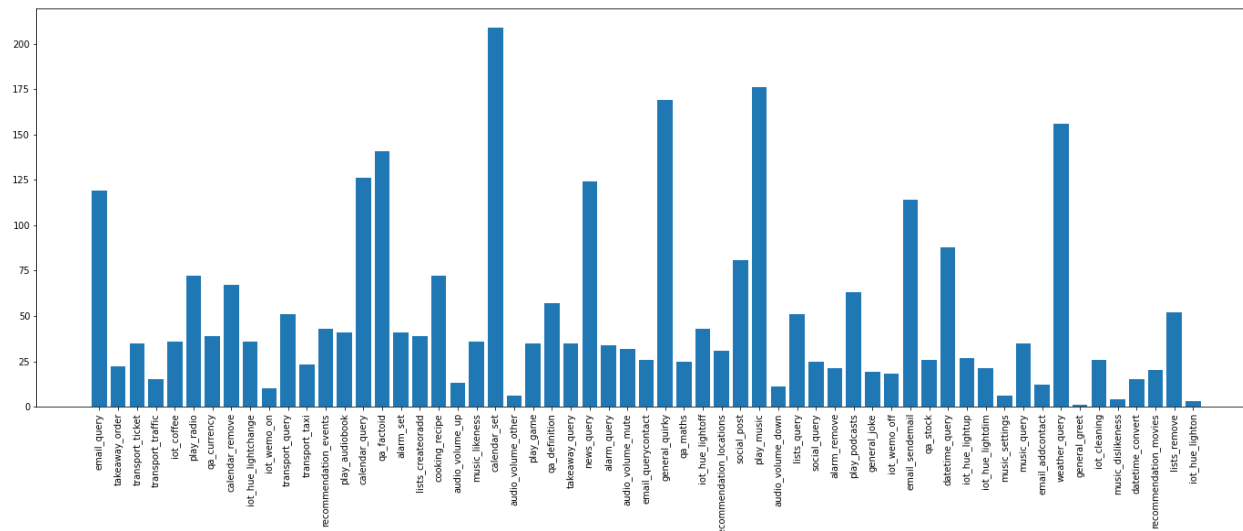


Fig5. Intents distribution in the test dataset

همانطور که مشاهده می‌شود شکل کلی توزیع داده‌ها در کلاس‌های مختلف intent در هر سه مجموعه داده آموزش، ارزیابی و تست یکسان است. می‌دانیم بالانس نبودن داده‌ها می‌تواند بر عملکرد مدل اثر گذار باشد. اگر چه در این دیتاست و در رابطه با intentها ، تعداد نمونه‌های برخی کلاس‌ها زیاد و برخی کلاس‌ها تعداد محدودی نمونه دارند اما به صورت متوسط می‌توان گفت داده‌ها تا حد قابل قبولی بالانس هستند.

- توزیع slotها در نمونه‌های کل دیتاست (آموزش، ارزیابی و تست روی هم):

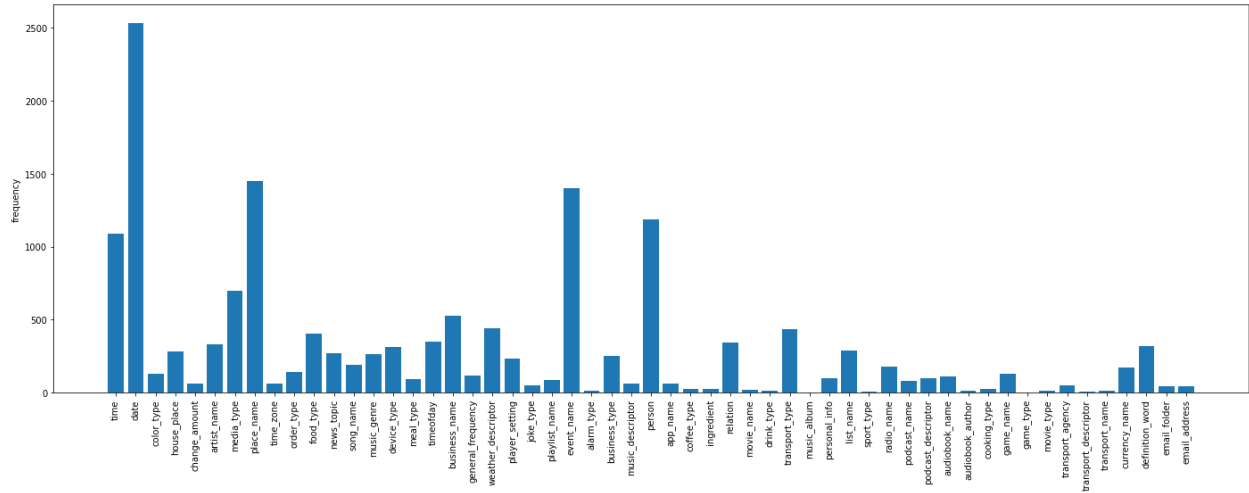


Fig6. Slots distribution in the whole dataset

- توزیع slotها در نمونه‌های دادگان آموزش:

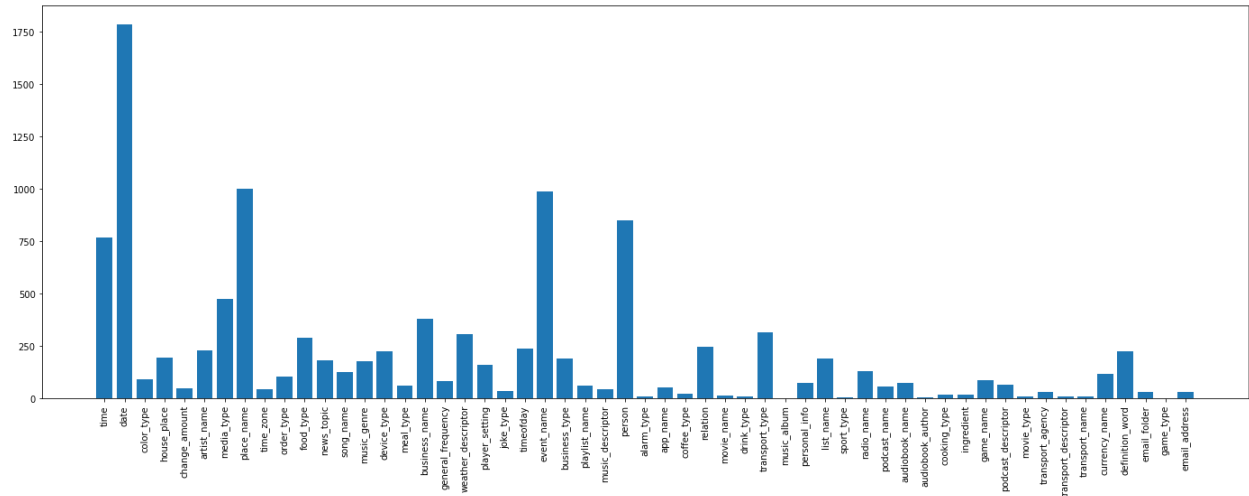


Fig7. Slots distribution in the train dataset

- توزیع slotها در نمونه‌های دادگان ارزیابی:

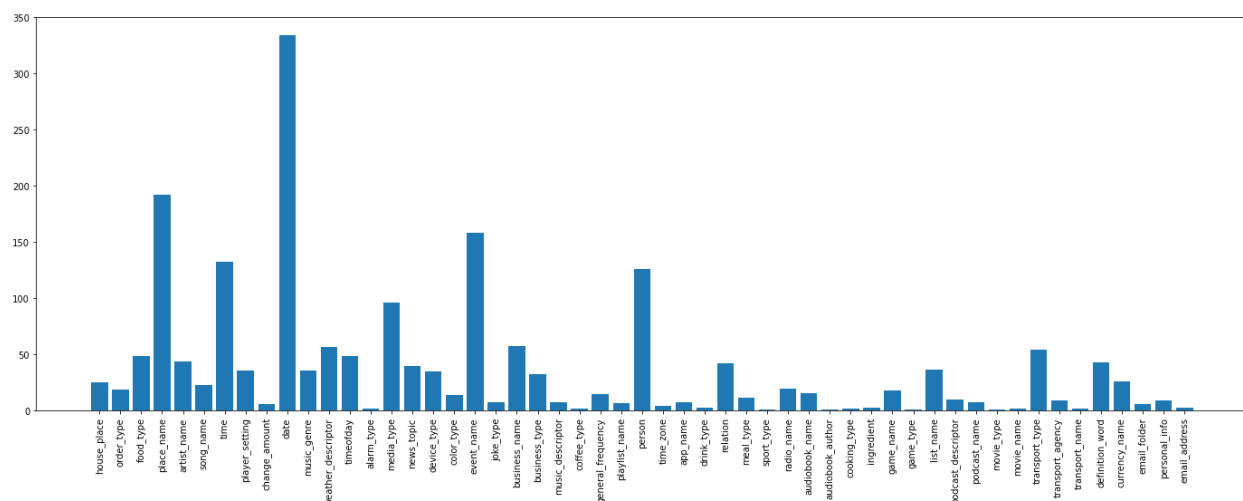


Fig8. Slots distribution in the eval dataset

● توزیع slotها در نمونه‌های دادگان تست:

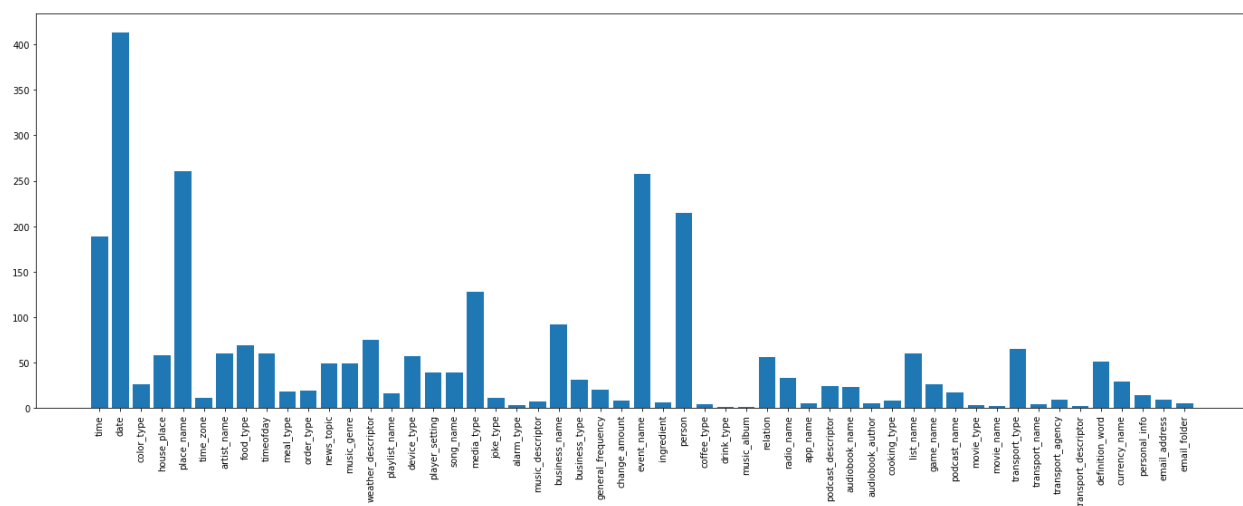


Fig9. Slots distribution in the test dataset

در مورد slotها مشاهده می‌شود که شکل کلی توزیع داده‌ها در کلاس‌های مختلف در مجموعه داده‌های آموزش، ارزیابی و تست کمتر شبیه به هم است و توزیع‌های نسبتاً متفاوتی را شاهد هستیم. همچنین مشاهده می‌شود که این بار توزیع داده‌ها کمتر متوازن است. به عبارت دیگر چند کلاس وجود دارد (نظیر time و date در مجموعه داده آموزش) که اکثریت نمونه‌ها به آن کلاس‌ها تعلق دارند و در مقابل بسیاری از کلاس‌های دیگر هستند که تعداد بسیار اندکی نمونه دارند. انتظار می‌رود این عدم توازن بر عملکرد مدل اثر بگذارد و احتمالاً دقت آن را کاهش دهد. البته لازم به ذکر است مدل‌های پیشرفته نظیر ترنسفورمرها توانایی یادگیری حتی با تعداد محدودی نمونه را هم دارند. در قسمت نتایج عملکرد این مدل‌ها در انجام این تسک قابل مشاهده است که با توجه به حجم نه چندان زیاد نمونه‌ها و بالانس نبودن آن‌ها عملکرد بسیار خوبی را شاهد بودیم.

Method

برای انجام این تسک از مدل‌های مبتنی بر ترنسفورمر چند زبانه یا multilingual استفاده می‌کنیم. این مدل‌ها در سال‌های اخیر در تسک‌های مختلف NLP، نتایج state-of-the-art از خود نشان داده‌اند. برای تسک NLU نیز می‌توان از این مدل‌ها استفاده کرد. یکی از روش‌های state-of-the-art برای این تسک، ایده ارائه شده در مقاله JointBERT است. (در مقاله MASSIVE نیز نتایج مدل‌هایی بر مبنای همین مقاله را به عنوان baseline بر روی دیتاست خود ارائه کرده‌اند). در این روش دو کلسیفایر روی خروجی BERT (یا هر مدل ترنسفورمر دیگر) قرار می‌گیرد. یکی از این کلسیفایرها برای تعیین intent است و دیگری برای تعیین slot‌ها. ما در این پروژه به دلیل اینکه دیتای فارسی داشتیم، از دو مدل چند زبانه که نتایج state-of-the-art داشتند یعنی XLM-RoBERTa و mT5 (هر دو مدل با سایز base که به ترتیب 270 میلیون و 258 میلیون پارامتر دارند) استفاده کردیم (همانند مقاله MASSIVE). برای هایپرپارامترها نیز از هایپرپارامترهای بهینه که در مقاله MASSIVE آمده بود استفاده کردیم زیرا در این مقاله با استفاده از 8 عدد GPU و طی چند روز هایپرپارامتر تیونینگ انجام شده که انجام این کار با توجه به منابع در دسترس ما در Google Colab عملاً امکان‌پذیر نبود بنابراین به هایپرپارامترهای بدست آمده در این مقاله بسنده کردیم هرچند که در این مقاله از کل دیتاست MASSIVE استفاده شده بود اما ما در این پروژه از یک زیر مجموعه که تنها شامل جملات فارسی بود استفاده کردیم و بنابراین ممکن است هایپرپارامترهای استفاده شده بهینه‌ترین نباشند و بتوان هایپرپارامترهای بهینه‌تری نیز پیدا کرد اما با توجه به محدودیت منابع امکان تیون کردن هایپر پارامترها برای ما فراهم نبود.

در مقاله JointBERT برای intent classifier از بردار متناظر با توکن CLS (اولین توکن در ترنسفورمرها) استفاده شده بود. اما در مقاله MASSIVE این مورد هم به عنوان یک هایپرپارامتر بررسی شده بود که به جای این بردار از میانگین یا ماکسیمم کل بردارهای خروجی ترنسفورمر استفاده کنیم؛ و بر این اساس مشاهده شده بود که در مدلی که بر مبنای ترنسفورمر mT5 بود همان استفاده از اولین توکن بهترین نتیجه را حاصل می‌کند اما در مدلی که بر مبنای XLM-RoBERTa بود استفاده از ماکسیمم گیری از بردارهای خروجی، بهترین نتیجه را می‌دهد که ما هم به همین منوال عمل کردیم. برای slot classifier هم دقیقاً مشابه مقاله JointBERT، تمام sequence خروجی ترنسفورمر را به کلسیفایر برای تعیین slot‌ها می‌دهیم. تفاوت دیگری که با مقاله JointBERT در پیاده‌سازی در اینجا وجود دارد این است که در آن مقاله، کلسیفایرها عملاً یک تبدیل خطی هستند که روی آنها softmax اعمال می‌شود اما در این پروژه برای بهبود عملکرد مدل با توجه به پیچیدگی بیشتر مدل‌های چند زبانه نسبت به مدل‌های تک زبانه استفاده شده در مقاله JointBERT نظیر BERT، از کلسیفایر غیر خطی (شبکه عصبی FeedForward چند لایه) با تابع فعال‌ساز GELU استفاده کردیم (همانند مقاله MASSIVE). همچنین برای آموزش این دو کلسیفایر به طور همزمان، loss مربوط به این دو کلسیفایر را با ضرایبی با هم جمع کردیم (مشابه مقاله JointBERT) که این ضرایب هم به عنوان هایپرپارامتر در مقاله MASSIVE بررسی شده بود که در این پروژه ما هم از همان مقدار بهینه استفاده کردیم. این ایده که به صورت توام و Joint این دو تسک را انجام دهیم و مدل Joint را آموزش دهیم، با توجه به نزدیکی ماهوی و مفهومی این دو تسک باعث شده است که این مدل یکی از مدل‌های state-of-the-art باشد.

مورد آخر اینکه در مقاله JointBERT حالتی را با اضافه کردن لایه CRF بعد از ترنسفورمر را هم بررسی کرده بود که با توجه به اینکه استفاده از آن تفاوت معناداری ایجاد نکرده بود در این پروژه نیز از این لایه استفاده نکردیم به خصوص که در این پروژه، دیتاست ما برای Slot filling از تگ‌های IOB استفاده نکرده است.

برای پیاده‌سازی موارد فوق از کد ارائه شده برای مقاله MASSIVE که در گیت‌هاب موجود است استفاده کردیم. البته نیاز به تغییراتی در کد ارائه شده بود تا با دیتاست ما که فقط فارسی بود سازگار باشد و همچنین قابلیت ادامه دادن روند آموزش از checkpoint نیز به آن اضافه شد تا وقتی محدودیت زمانی Colab تمام شد، بتوانیم آموزش را بعداً ادامه دهیم. کدهای موجود در گیت‌هاب MASSIVE هم بر اساس کدهای موجود در گیت‌هاب برای JointBERT بودند که با کمک Pytorch و کتابخانه

پر قدرت Huggingface Transformer پیاده شده بودند. برای توکنایز کردن ورودی نیز از Huggingface tokenizer استفاده کردیم که با توجه به سازگاری کامل با کتابخانه ترنسفورمر، با استفاده از آن به سادگی می‌توان داده‌های متنی را به طوری که قابلیت feed شدن به ترانسفورمر را دارد تبدیل کرد. پروسه توکنایز کردن و همچنین خروجی توکنایز شده کاملاً مشابه پارت اول پروژه و حتی ساده‌تر از آن است که چون این موضوع در پارت اول به تفصیل توضیح داده شده بود؛ بنابراین از تکرار مجدد این مورد در اینجا پرهیز می‌کنیم. در نهایت مدل‌ها را براساس توضیحات فوق روی دیتاست توضیح داده شده برای epoch 50 آموزش دادیم که نتایج آن در قسمت بعد قابل مشاهده است. مراحل آموزش برای جلوگیری از شلوغی و طولانی شدن بیش از حد گزارش در اینجا نیامده است اما در فایل نوت‌بوک ضمیمه شده مراحل آموزش هر کدام از مدل‌ها در دسترس است.

Results

mT5 based model

مقادیر هایپرپارامترهای اصلی مدل (که مطابق مقاله MASSIVE) تنظیم شده در ادامه قابل مشاهده است:

Table1. mT5 based model Hyperparamters

Model Hyperparameter	Value
d_ff	2048
d_kv	64
d_model	768
dropout_rate	0.1
feed_forward_proj	gated-gelu
initializer_factor	1.0
layer_norm_epsilon	1e-06
num_heads	12
num_layers	12
relative_attention_num_buckets	32
vocab_size	250112
use_crf	False
slot_loss_coef	4.0
hidden_dropout_prob	0.25

hidden_layer_for_class	9
head_num_layers	1
head_layer_dim	1024
attention_probs_dropout_prob	0.45
head_intent_pooling	First

Table2. mT5 based model optimization parameters

Optimizer Parameter	Value
learning_rate	3.525e-4
lr_scheduler_type	constant_with_warmup
warmup_steps	600
adam_beta1	0.8
adam_beta2	0.999
adam_epsilon	1.0e-9
weight_decay	0.07
gradient_accumulation_steps	8
per_device_train_batch_size	32
per_device_eval_batch_size	64
num_train_epochs	50

در شکل‌های 9 الی 11 به ترتیب نمودارهای F1-score (روی دیتای ارزیابی) برای تسک slot filling و Accuracy (روی دیتای ارزیابی) برای تسک intent classification و loss در stepهای مختلف آموزش مدل آمده است. مشاهده می‌شود که با افزایش stepهای آموزش از یک جایی به بعد F1 و Accuracy ثابت شده و حتی شاید اندکی کاهش یافته است که نشان می‌دهد مدل دیگر الگوی جدیدی از داده‌ها یاد نمی‌گیرد و آموزش بیش از این ممکن است موجب overfit شدن روی داده آموزش و کاهش دقت مدل روی داده ارزیابی و تست شود به همین دلیل بیش از 50 تا epoch آموزش مدل را ادامه ندادیم.

eval/fa-IR_slot_micro_f1
tag: eval/fa-IR_slot_micro_f1

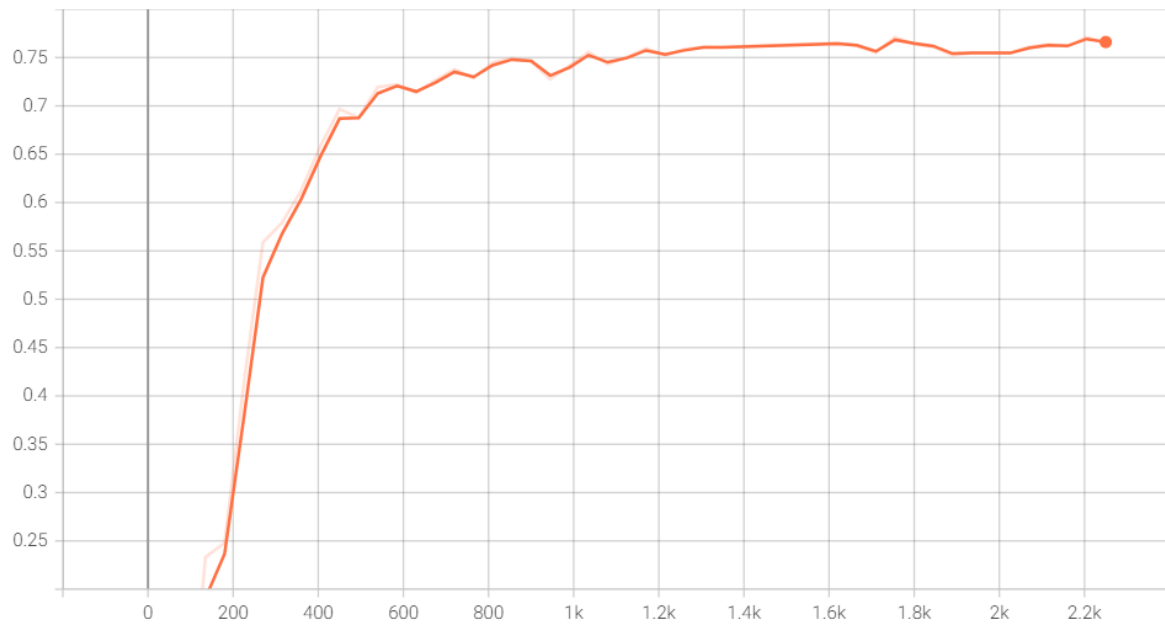


Fig9. mT5 based model slot micro avg f1 plot

eval/fa-IR_intent_acc
tag: eval/fa-IR_intent_acc

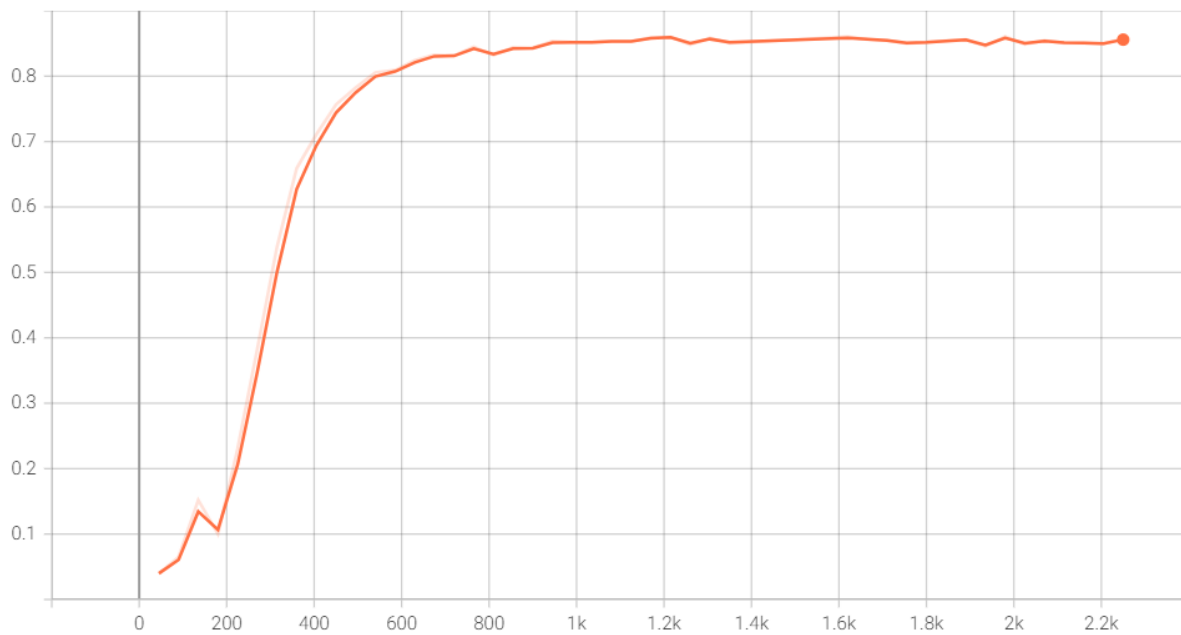


Fig10. mT5 based model intent accuracy plot

train/train_loss
tag: train/train_loss

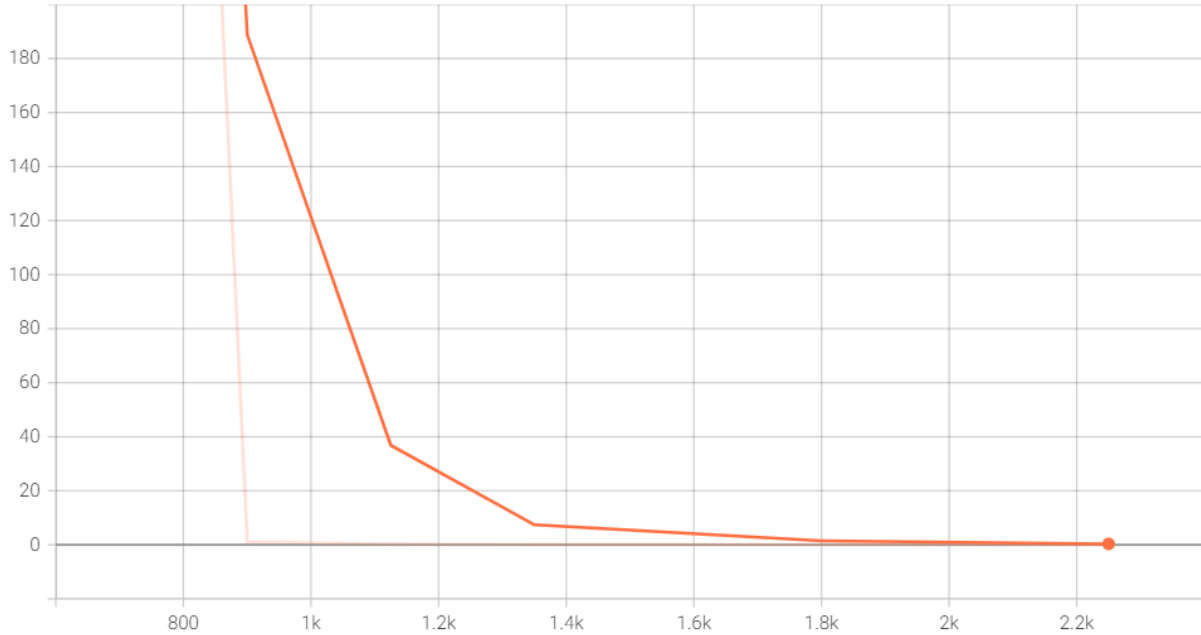


Fig11. mT5 based model loss plot

نتایج نهایی بهترین مدل (بدست آمده از checkpoint در آخرین step آموزش) روی مجموعه داده تست:

Table3. mT5 based model results on test dataset

Exact Match Accuracy	64.5 ± 1.8
Micro-Avg Slot F1	75.3 ± 0.15
Intent Accuracy	85.9 ± 1.3

چند نمونه از خروجی مدل:

```
{
  "id": "12446",
  "locale": "fa-IR",
  "utt": ["", "به", "زمانی", "چه", "قطار", "که", "بگویی", "من", "به", "می‌توانی"],
  "pred_intent": "transport_query",
  "pred_slots": ["قطار", "transport_type", "مشهد", "place_name"],
  "pred_annot_utt": "می‌توانی به من بگویی که [transport_type : مشهد] [place_name : قطار] چه زمانی به سمت [مشهد] حرکت می‌کند"
},
{
  "id": "9568",
  "locale": "fa-IR",
  "utt": ["کن", "پخش", "را", "ابی", "زنده", "اجراهای", "مجموعه"],
  "pred_intent": "play_music",
  "pred_slots": ["اجراهای زنده", "music_descriptor", "ابی", "artist_name"],
  "pred_annot_utt": "ابی [را : مجموعه] [music_descriptor : اجراهای زنده] [artist_name : پخش کن]"
}
```

{ "id": "6923", "locale": "fa-IR", "utt": [" ", "شنبه", "سه", "صبح", "در", "اسناد", "چاپ", "برای", "یادآور", "یک", "یادآور", "برای", "چاپ", "اسناد", "در", "صبح", "تنظیم", "کن", "چاپ اسناد", "در : یک یادآور برای", "pred_annot_utt": " [timeofday : صبح [date : سه شنبه]]", "pred_intent": "calendar_set", "pred_slots": [["event_name", "چاپ اسناد"], ["timeofday", "سه شنبه", "date"]], "event_name": "چاپ اسناد", "timeofday": "صبح", "date": "سه شنبه" }

XLM-RoBERTa based model

مقادیر هایپرپارامترهای اصلی مدل (که مطابق مقاله MASSIVE) تنظیم شده در ادامه قابل مشاهده است:

Table4. XLM-RoBERTa based model hyperparameters

Model HyperParameter	Value
attention_probs_dropout_prob	0.0
hidden_act	gelu
hidden_dropout_prob	0.45
hidden_size	768
initializer_range	0.02
intermediate_size	3072
layer_norm_eps	1e-5
max_position_embeddings	514
num_attention_heads	12
num_hidden_layers	12
vocab_size	25002
use_crf	False
slot_loss_coef	4.0
hidden_layer_for_class	11
head_num_layers	1
head_layer_dim	2048
head_intent_pooling	Max

Table5. XLM-RoBERTa based model optimization parameters

Optimizer Parameter	Value
learning_rate	2.8e-05
lr_scheduler_type	constant_with_warmup
warmup_steps	800
adam_beta1	0.9
adam_beta2	0.9999
adam_epsilon	1.0e-08
weight_decay	0.21
gradient_accumulation_steps	1
per_device_train_batch_size	128
per_device_eval_batch_size	128
num_train_epochs	50

در شکل‌های 12 الی 14 به ترتیب نمودارهای F1-score (روی دیتای ارزیابی) برای تسک slot filling و Accuracy (روی دیتای ارزیابی) برای تسک intent classification و loss در stepهای مختلف آموزش مدل آمده است. مشابه مدل قبلی اینجا هم مشاهده می‌شود که با افزایش stepهای آموزش از یک جایی به بعد F1 و Accuracy ثابت شده و حتی شاید اندکی کاهش یافته است بنابراین برای جلوگیری از overfitting این مدل را هم بیش از 50 تا epoch آموزش ندادیم.

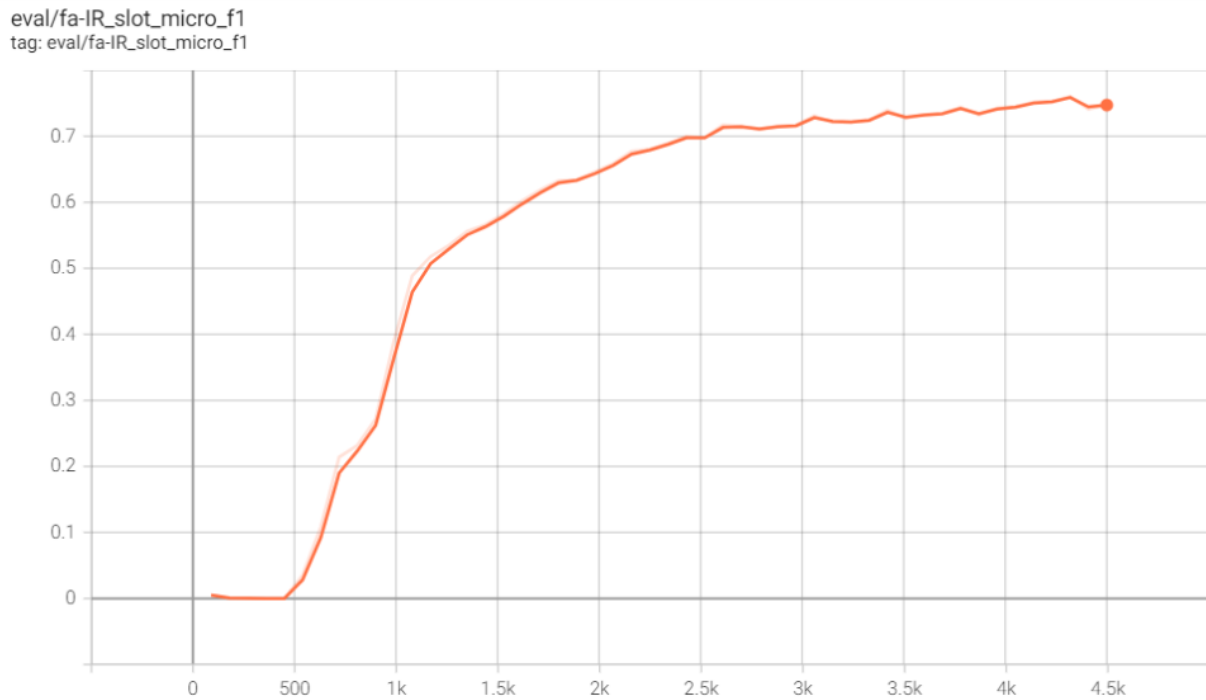


Fig12. XLM-RoBERTa based model slot micro avg f1 plot

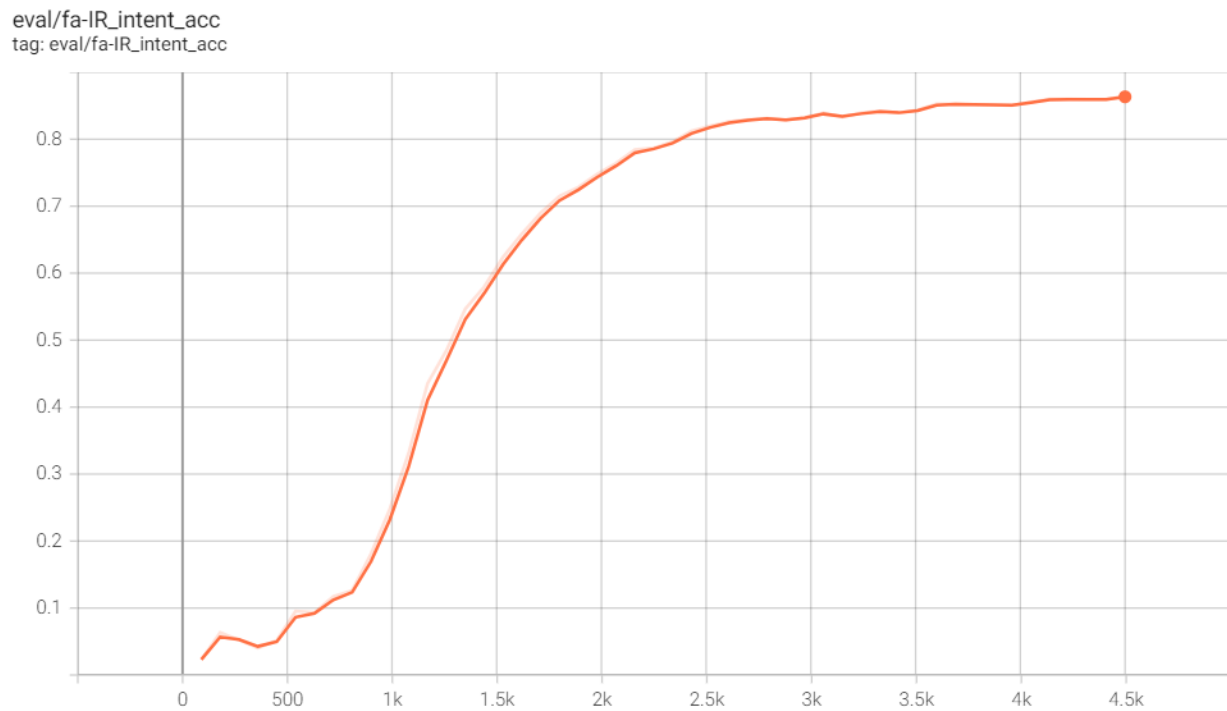


Fig13. XLM-RoBERTa based model intent accuracy plot

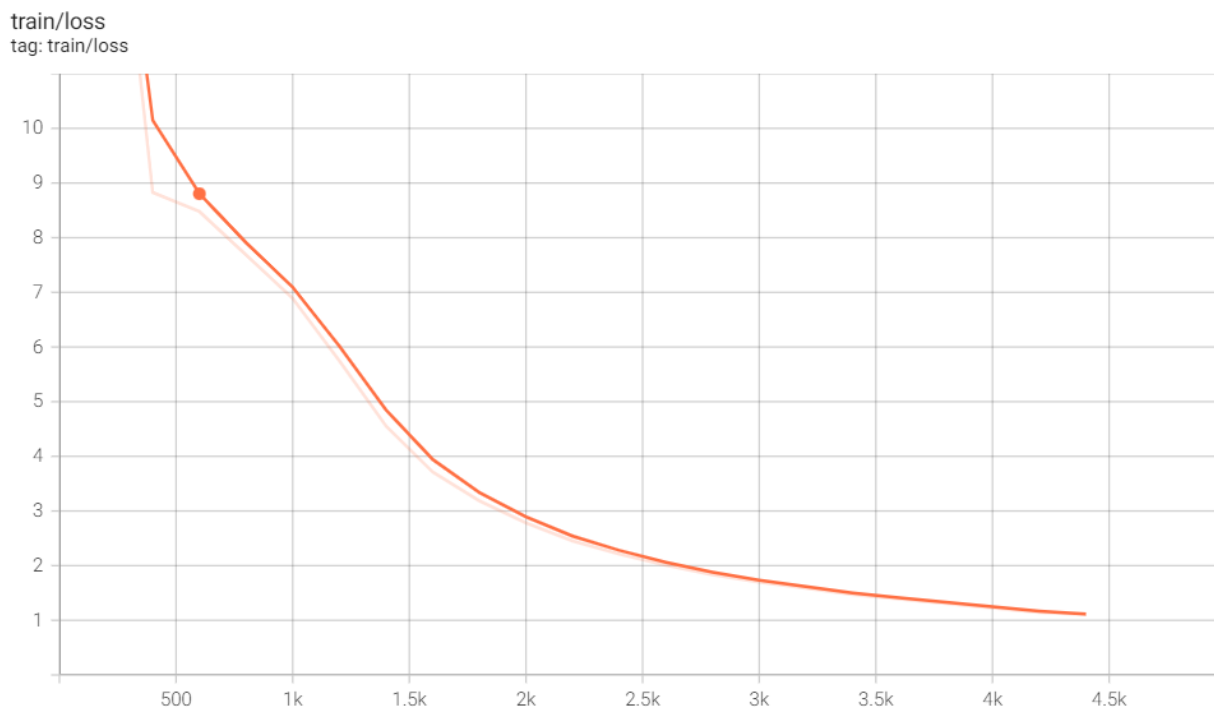


Fig14. XLM-RoBERTa based model loss plot

نتایج نهایی بهترین مدل (بدست آمده از checkpoint در آخرین step آموزش) روی مجموعه داده تست:

Table6. XLM-RoBERTa based model results on test dataset

Exact Match Accuracy	63.6 ± 1.8
Micro-Avg Slot F1	74.6 ± 0.2
Intent Accuracy	85.8 ± 1.3

چند نمونه از خروجی مدل:

```
{
  "id": "12446",
  "locale": "fa-IR",
  "utt": ["", "به", "زمانی", "چه", "قطار", "که", "بگویی", "من", "به", "می‌توانی"],
  "pred_intent": "transport_query",
  "pred_slots": ["قطار", "transport_type", ["مشهد", "place_name"]],
  "pred_annot_utt": "می‌توانی به من بگویی که [مشهد] حرکت می‌کند [place_name : چه زمانی به سمت : transport_type]",
  "id": "9568",
  "locale": "fa-IR",
  "utt": ["کن", "پخش", "را", "ابی", "زنده", "اجراهای", "مجموعه"],
  "pred_intent": "play_music",
  "pred_slots": ["مجموعه اجراهای زنده", "playlist_name", ["ابی", "artist_name"]],
  "pred_annot_utt": "ابی [را پخش [artist_name : مجموعه اجراهای زنده : playlist_name]] [کن]"
}
```


Conclusion

در پارت دوم این پروژه یک مدل state-of-the-art بر مبنای ترنسفورمرهای چند زبانه XLM-RoBERTa و mT5 برای تسک NLU و به طور خاص برای انجام Joint دو تسک intent classification و slot filling (که ارتباط تنگاتنگی با هم دارند) استفاده و آن‌ها را روی دیتای فارسی دیتاست MASSIVE فاین تیون کردیم و به نتایج مشابه مدل‌های state-of-the-art رسیدیم؛ حال اینکه از نظر منابع سخت افزاری و نیز زمانی محدود بودیم که این نشان از قدرت بالای representation‌های ارائه شده توسط ترنسفورمرها و نیز قدرت آن‌ها در فاین تیون شدن با داده کم و در زمان کم (تعداد epoch کم) است. هر چند طبق مشاهدات ما همین مقداری که مدل‌ها را آموزش دادیم کافی بوده زیرا بیشتر از آن مدل‌ها به سمت overfit شدن پیش می‌رفتند. نکته ای که لازم به ذکر است این است که دیتاست ما چندان متوازن نبود، بنابراین در پروژه‌های آینده می‌توان این موضوع را با روش‌هایی نظیر undersampling (مثلاً حذف جملات و نمونه‌های مشابه از کلاس‌های با تعداد عضو بالا) و oversampling (تولید نمونه برای کلاس‌های با تعداد عضو کم) و... تا حد امکان برطرف کرد؛ با بهبود توازن دیتاست انتظار می‌رود عملکرد مدل‌ها نیز بهبود پیدا کند. همچنین در کارهای بعدی می‌توان بررسی کرد که فاین تیون کردن مدل با دیتا زبانی که از نظر ساختار زبانی (زبان‌های هندواروپایی مثل انگلیسی) یا از نظر رسم الخط (نظیر عربی) با فارسی مشابه هستند، می‌توان عملکرد مدل را روی داده تست فارسی بهبود بخشید یا خیر.

References

- [1] Qian Chen, Zhu Zhuo, Wen Wang: “BERT for Joint Intent Classification and Slot Filling”, 2019; arXiv:1902.10909.
- [2] Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, Prem Natarajan: “MASSIVE: A 1M-Example Multilingual Natural Language Understanding Dataset with 51 Typologically-Diverse Languages”, 2022; arXiv:2204.08582.
- [3] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.
- [4] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer.
- [5] Dan Hendrycks, Kevin Gimpel: “Gaussian Error Linear Units (GELUs)”, 2016; arXiv:1606.08415.

Appendix

Intents distribution (Train+Eval+Test):

takeaway_order: 177 (1.07%),
transport_traffic: 154 (0.93%),
qa_factoid: 775 (4.69%),
weather_query: 855 (5.18%),
iot_hue_lightup: 115 (0.70%),
play_podcasts: 290 (1.76%),
audio_volume_up: 135 (0.82%),
transport_ticket: 187 (1.13%),
email_querycontact: 169 (1.02%),
iot_hue_lightdim: 114 (0.69%),
transport_taxi: 150 (0.91%),
alarm_query: 183 (1.11%),
lists_remove: 253 (1.53%),
email_query: 610 (3.69%),
recommendation_events: 259 (1.57%),
iot_coffee: 174 (1.05%),
iot_hue_lightchange: 183 (1.11%),
play_audiobook: 226 (1.37%),
play_game: 169 (1.02%),
calendar_set: 1150 (6.96%),
recommendation_movies: 102 (0.62%),
cooking_query: 6 (0.04%),
datetime_convert: 76 (0.46%),
takeaway_query: 181 (1.10%),
general_greet: 28 (0.17%),
iot_cleaning: 138 (0.84%),
alarm_set: 254 (1.54%),
calendar_query: 794 (4.81%),
music_dislikeness: 20 (0.12%),
qa_stock: 202 (1.22%),
music_likeness: 165 (1.00%),
social_post: 414 (2.51%),
social_query: 151 (0.91%),
alarm_remove: 113 (0.68%),
general_joke: 106 (0.64%),
lists_createoradd: 241 (1.46%),
qa_currency: 213 (1.29%),
iot_hue_lighton: 30 (0.18%),
audio_volume_down: 71 (0.43%),
recommendation_locations: 235 (1.42%),
play_music: 938 (5.68%),
cooking_recipe: 320 (1.94%),
qa_definition: 379 (2.29%),
iot_wemo_off: 75 (0.45%),
audio_volume_other: 24 (0.15%),
music_query: 219 (1.33%),
email_addcontact: 71 (0.43%),
music_settings: 65 (0.39%),
email_sendemail: 531 (3.21%),
datetime_query: 502 (3.04%),

play_radio: 401 (2.43%),
qa_maths: 116 (0.70%),
transport_query: 314 (1.90%),
lists_query: 299 (1.81%),
calendar_remove: 426 (2.58%),
iot_hue_lightoff: 213 (1.29%),
news_query: 709 (4.29%),
iot_wemo_on: 65 (0.39%),
audio_volume_mute: 157 (0.95%),
general_quirky: 829 (5.02%),

Slots distribution (Train+Eval+Test):

time: 1091 (6.95%),
date: 2531 (16.13%),
color_type: 133 (0.85%),
house_place: 280 (1.78%),
change_amount: 63 (0.40%),
artist_name: 332 (2.12%),
media_type: 698 (4.45%),
place_name: 1452 (9.25%),
time_zone: 60 (0.38%),
order_type: 144 (0.92%),
food_type: 407 (2.59%),
news_topic: 271 (1.73%),
song_name: 189 (1.20%),
music_genre: 265 (1.69%),
device_type: 316 (2.01%),
meal_type: 93 (0.59%),
timeofday: 348 (2.22%),
business_name: 529 (3.37%),
general_frequency: 119 (0.76%),
weather_descriptor: 440 (2.80%),
player_setting: 234 (1.49%),
joke_type: 53 (0.34%),
playlist_name: 85 (0.54%),
event_name: 1404 (8.95%),
alarm_type: 14 (0.09%),
business_type: 253 (1.61%),
music_descriptor: 60 (0.38%),
person: 1189 (7.58%),
app_name: 64 (0.41%),
coffee_type: 28 (0.18%),
ingredient: 27 (0.17%),
relation: 344 (2.19%),
movie_name: 20 (0.13%),
drink_type: 13 (0.08%),
transport_type: 435 (2.77%),
music_album: 2 (0.01%),
personal_info: 97 (0.62%),
list_name: 287 (1.83%),
sport_type: 6 (0.04%),
radio_name: 182 (1.16%),
podcast_name: 81 (0.52%),

podcast_descriptor: 101 (0.64%),
audiobook_name: 112 (0.71%),
audiobook_author: 12 (0.08%),
cooking_type: 27 (0.17%),
game_name: 131 (0.83%),
game_type: 2 (0.01%),
movie_type: 14 (0.09%),
transport_agency: 49 (0.31%),
transport_descriptor: 10 (0.06%),
transport_name: 15 (0.10%),
currency_name: 173 (1.10%),
definition_word: 319 (2.03%),
email_folder: 43 (0.27%),
email_address: 43 (0.27%),

Intents distribution (Train):

takeaway_order: 135 (1.17%),
transport_traffic: 117 (1.02%),
qa_factoid: 544 (4.72%),
weather_query: 573 (4.98%),
iot_hue_lightup: 76 (0.66%),
play_podcasts: 193 (1.68%),
audio_volume_up: 110 (0.96%),
transport_ticket: 127 (1.10%),
email_querycontact: 127 (1.10%),
iot_hue_lightdim: 76 (0.66%),
transport_taxi: 100 (0.87%),
alarm_query: 130 (1.13%),
lists_remove: 164 (1.42%),
email_query: 418 (3.63%),
recommendation_events: 190 (1.65%),
iot_coffee: 124 (1.08%),
iot_hue_lightchange: 125 (1.09%),
play_audiobook: 150 (1.30%),
play_game: 112 (0.97%),
calendar_set: 810 (7.03%),
recommendation_movies: 70 (0.61%),
cooking_query: 4 (0.03%),
datetime_convert: 52 (0.45%),
takeaway_query: 122 (1.06%),
general_greet: 25 (0.22%),
iot_cleaning: 93 (0.81%),
alarm_set: 182 (1.58%),
calendar_query: 566 (4.92%),
music_dislikeness: 14 (0.12%),
qa_stock: 152 (1.32%),
music_likeness: 113 (0.98%),
social_post: 283 (2.46%),
social_query: 108 (0.94%),
alarm_remove: 78 (0.68%),
general_joke: 72 (0.63%),
lists_createoradd: 177 (1.54%),

qa_currency: 142 (1.23%),
 iot_hue_lighton: 22 (0.19%),
 audio_volume_down: 52 (0.45%),
 recommendation_locations: 173 (1.50%),
 play_music: 639 (5.55%),
 qa_definition: 267 (2.32%),
 cooking_recipe: 207 (1.80%),
 iot_wemo_off: 52 (0.45%),
 audio_volume_other: 18 (0.16%),
 music_query: 154 (1.34%),
 email_addcontact: 54 (0.47%),
 music_settings: 51 (0.44%),
 email_sendemail: 354 (3.07%),
 datetime_query: 350 (3.04%),
 play_radio: 283 (2.46%),
 qa_maths: 78 (0.68%),
 transport_query: 227 (1.97%),
 lists_query: 198 (1.72%),
 calendar_remove: 312 (2.71%),
 iot_hue_lightoff: 153 (1.33%),
 news_query: 503 (4.37%),
 iot_wemo_on: 48 (0.42%),
 audio_volume_mute: 110 (0.96%),
 general_quirky: 555 (4.82%),

Slots distribution (Train):

time: 769 (6.99%),
 date: 1784 (16.22%),
 color_type: 93 (0.85%),
 house_place: 197 (1.79%),
 change_amount: 49 (0.45%),
 artist_name: 228 (2.07%),
 media_type: 474 (4.31%),
 place_name: 999 (9.08%),
 time_zone: 45 (0.41%),
 order_type: 106 (0.96%),
 food_type: 289 (2.63%),
 news_topic: 182 (1.65%),
 song_name: 127 (1.15%),
 music_genre: 180 (1.64%),
 device_type: 224 (2.04%),
 meal_type: 63 (0.57%),
 business_name: 379 (3.45%),
 general_frequency: 84 (0.76%),
 weather_descriptor: 308 (2.80%),
 player_setting: 159 (1.45%),
 joke_type: 34 (0.31%),
 timeofday: 239 (2.17%),
 event_name: 988 (8.98%),
 business_type: 189 (1.72%),
 playlist_name: 62 (0.56%),
 music_descriptor: 45 (0.41%),
 person: 848 (7.71%),

alarm_type: 9 (0.08%),
 app_name: 51 (0.46%),
 coffee_type: 22 (0.20%),
 relation: 246 (2.24%),
 movie_name: 16 (0.15%),
 drink_type: 9 (0.08%),
 transport_type: 316 (2.87%),
 music_album: 1 (0.01%),
 personal_info: 74 (0.67%),
 list_name: 190 (1.73%),
 sport_type: 5 (0.05%),
 radio_name: 129 (1.17%),
 podcast_name: 56 (0.51%),
 audiobook_name: 73 (0.66%),
 audiobook_author: 6 (0.05%),
 cooking_type: 17 (0.15%),
 ingredient: 18 (0.16%),
 game_name: 87 (0.79%),
 podcast_descriptor: 67 (0.61%),
 movie_type: 10 (0.09%),
 transport_agency: 31 (0.28%),
 transport_descriptor: 8 (0.07%),
 transport_name: 9 (0.08%),
 currency_name: 118 (1.07%),
 definition_word: 225 (2.05%),
 email_folder: 32 (0.29%),
 game_type: 1 (0.01%),
 email_address: 31 (0.28%),

Intents distribution (Eval):

takeaway_order: 20 (0.98%),
 general_quirky: 105 (5.16%),
 transport_traffic: 22 (1.08%),
 qa_factoid: 90 (4.43%),
 weather_query: 126 (6.20%),
 iot_hue_lightup: 12 (0.59%),
 play_podcasts: 34 (1.67%),
 audio_volume_up: 12 (0.59%),
 transport_ticket: 25 (1.23%),
 transport_taxi: 27 (1.33%),
 iot_hue_lightdim: 17 (0.84%),
 email_querycontact: 16 (0.79%),
 alarm_query: 19 (0.93%),
 lists_remove: 37 (1.82%),
 email_query: 73 (3.59%),
 recommendation_events: 26 (1.28%),
 iot_coffee: 14 (0.69%),
 iot_hue_lightchange: 22 (1.08%),
 play_audiobook: 35 (1.72%),
 play_game: 22 (1.08%),
 calendar_set: 131 (6.44%),
 recommendation_movies: 12 (0.59%),
 cooking_query: 2 (0.10%),

datetime_convert: 9 (0.44%),
 takeaway_query: 24 (1.18%),
 general_greet: 2 (0.10%),
 iot_cleaning: 19 (0.93%),
 music_dislikeness: 2 (0.10%),
 calendar_query: 102 (5.02%),
 qa_stock: 24 (1.18%),
 music_likeness: 16 (0.79%),
 social_post: 50 (2.46%),
 social_query: 18 (0.89%),
 alarm_remove: 14 (0.69%),
 general_joke: 15 (0.74%),
 lists_createoradd: 25 (1.23%),
 qa_currency: 32 (1.57%),
 iot_hue_lighton: 5 (0.25%),
 audio_volume_down: 8 (0.39%),
 cooking_recipe: 41 (2.02%),
 play_music: 123 (6.05%),
 recommendation_locations: 31 (1.52%),
 qa_definition: 55 (2.71%),
 iot_wemo_off: 5 (0.25%),
 email_addcontact: 5 (0.25%),
 music_query: 30 (1.48%),
 music_settings: 8 (0.39%),
 email_sendemail: 63 (3.10%),
 datetime_query: 64 (3.15%),
 play_radio: 46 (2.26%),
 qa_maths: 13 (0.64%),
 transport_query: 36 (1.77%),
 lists_query: 50 (2.46%),
 calendar_remove: 47 (2.31%),
 iot_hue_lightoff: 17 (0.84%),
 news_query: 82 (4.03%),
 iot_wemo_on: 7 (0.34%),
 audio_volume_mute: 15 (0.74%),
 alarm_set: 31 (1.52%),

Slots distribution (Eval):

house_place: 25 (1.28%),
 order_type: 19 (0.98%),
 food_type: 49 (2.52%),
 place_name: 192 (9.87%),
 artist_name: 44 (2.26%),
 song_name: 23 (1.18%),
 time: 133 (6.83%),
 player_setting: 36 (1.85%),
 change_amount: 6 (0.31%),
 date: 334 (17.16%),
 music_genre: 36 (1.85%),
 weather_descriptor: 57 (2.93%),
 timeofday: 49 (2.52%),
 alarm_type: 2 (0.10%),
 media_type: 96 (4.93%),

news_topic: 40 (2.06%),
device_type: 35 (1.80%),
color_type: 14 (0.72%),
event_name: 158 (8.12%),
joke_type: 8 (0.41%),
business_name: 58 (2.98%),
business_type: 33 (1.70%),
music_descriptor: 8 (0.41%),
coffee_type: 2 (0.10%),
general_frequency: 15 (0.77%),
playlist_name: 7 (0.36%),
person: 126 (6.47%),
time_zone: 4 (0.21%),
app_name: 8 (0.41%),
drink_type: 3 (0.15%),
relation: 42 (2.16%),
meal_type: 12 (0.62%),
sport_type: 1 (0.05%),
radio_name: 20 (1.03%),
audiobook_name: 16 (0.82%),
audiobook_author: 1 (0.05%),
cooking_type: 2 (0.10%),
ingredient: 3 (0.15%),
game_name: 18 (0.92%),
game_type: 1 (0.05%),
list_name: 37 (1.90%),
podcast_descriptor: 10 (0.51%),
podcast_name: 8 (0.41%),
movie_type: 1 (0.05%),
movie_name: 2 (0.10%),
transport_type: 54 (2.77%),
transport_agency: 9 (0.46%),
transport_name: 2 (0.10%),
definition_word: 43 (2.21%),
currency_name: 26 (1.34%),
email_folder: 6 (0.31%),
personal_info: 9 (0.46%),
email_address: 3 (0.15%),

Intents distribution (Test):

takeaway_order: 22 (0.74%),
transport_traffic: 15 (0.50%),
qa_factoid: 141 (4.74%),
weather_query: 156 (5.25%),
iot_hue_lightup: 27 (0.91%),
play_podcasts: 63 (2.12%),
audio_volume_up: 13 (0.44%),
transport_ticket: 35 (1.18%),
transport_taxi: 23 (0.77%),
iot_hue_lightdim: 21 (0.71%),
email_querycontact: 26 (0.87%),
alarm_query: 34 (1.14%),
lists_remove: 52 (1.75%),

email_query: 119 (4.00%),
 recommendation_events: 43 (1.45%),
 iot_coffee: 36 (1.21%),
 iot_hue_lightchange: 36 (1.21%),
 play_audiobook: 41 (1.38%),
 play_game: 35 (1.18%),
 calendar_set: 209 (7.03%),
 recommendation_movies: 20 (0.67%),
 datetime_convert: 15 (0.50%),
 takeaway_query: 35 (1.18%),
 general_greet: 1 (0.03%),
 iot_cleaning: 26 (0.87%),
 alarm_set: 41 (1.38%),
 music_dislikeness: 4 (0.13%),
 calendar_query: 126 (4.24%),
 qa_stock: 26 (0.87%),
 music_likeness: 36 (1.21%),
 social_post: 81 (2.72%),
 social_query: 25 (0.84%),
 alarm_remove: 21 (0.71%),
 general_joke: 19 (0.64%),
 lists_createoradd: 39 (1.31%),
 qa_currency: 39 (1.31%),
 iot_hue_lighton: 3 (0.10%),
 audio_volume_down: 11 (0.37%),
 recommendation_locations: 31 (1.04%),
 play_music: 176 (5.92%),
 cooking_recipe: 72 (2.42%),
 qa_definition: 57 (1.92%),
 iot_wemo_off: 18 (0.61%),
 audio_volume_other: 6 (0.20%),
 music_query: 35 (1.18%),
 email_addcontact: 12 (0.40%),
 music_settings: 6 (0.20%),
 email_sendemail: 114 (3.83%),
 datetime_query: 88 (2.96%),
 play_radio: 72 (2.42%),
 qa_maths: 25 (0.84%),
 transport_query: 51 (1.71%),
 lists_query: 51 (1.71%),
 calendar_remove: 67 (2.25%),
 iot_hue_lightoff: 43 (1.45%),
 news_query: 124 (4.17%),
 iot_wemo_on: 10 (0.34%),
 audio_volume_mute: 32 (1.08%),
 general_quirky: 169 (5.68%),

Slots distribution (Test):

time: 189 (6.89%),
 date: 413 (15.06%),
 color_type: 26 (0.95%),
 house_place: 58 (2.11%),
 place_name: 261 (9.52%),

time_zone: 11 (0.40%),
artist_name: 60 (2.19%),
food_type: 69 (2.52%),
timeofday: 60 (2.19%),
meal_type: 18 (0.66%),
order_type: 19 (0.69%),
news_topic: 49 (1.79%),
music_genre: 49 (1.79%),
weather_descriptor: 75 (2.73%),
playlist_name: 16 (0.58%),
device_type: 57 (2.08%),
player_setting: 39 (1.42%),
song_name: 39 (1.42%),
media_type: 128 (4.67%),
joke_type: 11 (0.40%),
alarm_type: 3 (0.11%),
music_descriptor: 7 (0.26%),
business_name: 92 (3.35%),
business_type: 31 (1.13%),
general_frequency: 20 (0.73%),
change_amount: 8 (0.29%),
event_name: 258 (9.41%),
ingredient: 6 (0.22%),
person: 215 (7.84%),
coffee_type: 4 (0.15%),
drink_type: 1 (0.04%),
music_album: 1 (0.04%),
relation: 56 (2.04%),
radio_name: 33 (1.20%),
app_name: 5 (0.18%),
podcast_descriptor: 24 (0.87%),
audiobook_name: 23 (0.84%),
audiobook_author: 5 (0.18%),
cooking_type: 8 (0.29%),
list_name: 60 (2.19%),
game_name: 26 (0.95%),
podcast_name: 17 (0.62%),
movie_type: 3 (0.11%),
movie_name: 2 (0.07%),
transport_type: 65 (2.37%),
transport_name: 4 (0.15%),
transport_agency: 9 (0.33%),
transport_descriptor: 2 (0.07%),
definition_word: 51 (1.86%),
currency_name: 29 (1.06%),
personal_info: 14 (0.51%),
email_address: 9 (0.33%),
email_folder: 5 (0.18%),