دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر

گزارش تمرین شماره ۵

درس NLP

بهار ۱۴۰۲

نام و نام خانوادگی
سپهر کریمی آرپناهی

شماره دانشجویی
۸۱۰۱۰۰۴۴۷

سوال اول

در این سوال می خواهیم تاثیر BPE را در سیستم ترجمه ماشینی بسنجیم برای این کار یک بار مدل را بدون BPE و یک بار با BPE اجرا می کنیم و در نهایت نمودار ها و نتایج بدست آمده را رسم می کنیم.

بخش اول : مدل بدون استفاده از BPE

در این قسمت ابتدا دادگان را در کولب load کرده و کتابخانه های مورد نیاز را فراخوانی می کنیم.

سپس مراحل زیر را به ترتیب دنبال می کنیم:

پیش پردازش دادگان با استفاده از fariseq-preprocess

همانطور که در hands-on گفته شده برای استفاده از fair-seq ابتدا باید از دستور fairseq-preprocess استفاده کرد. این دستور دیکشنری های fairseq را تولید می کند و داده ها را binarize می کند. در این دستور داده های آموزش، اعتبارسنجی و تست به عنوان ورودی داده می شود و مسیر مشخص شده data_bin مسیر ذخیره خروجی می باشد همچنین با استفاده از دستورات nwordssrc و nwordstgt اندازه vocab را برابر با ۴۰۰۰۰ تعیین می کنیم.

```
  --nwordssrc 40000 \
  --nwordstgt 40000
```

آموزش مدل با استفاده از fairseq-train

پس از ذخیره داده های binarize شده، حال می خواهیم مدل را آموزش دهیم. در این قسمت ابتدا خروجی قسمت قبل را که در محل data_bin ذخیره کردیم به عنوان ورودی اول می دهیم. سپس در قسمت arch-- معماری مدل را برابر با LSTM با مکانیزم attention قرار می دهیم. سپس بهینه ساز را برابر با adam قرار داده و پارامتر های beta1=0.9 و beta2=0.98 قرار می دهیم .همینطور نرخ یادگیری را با استفاده از lr 25e-4-- مقدار دهی اولیه می کنیم و با استفاده از lr-scheduler inverse_sqrt-- آن را تطبیقی می کنیم. در ادامه مقدار dropout را برابر با ۰.۲۵ قرار می دهیم. سپس تابع هزینه را برابر با label smoothed cross entropy قرار داده و مقدار label smoothing را برابر با 0.2 قرار می دهیم. همچنین تعداد ایپاک را برابر با ۶ قرار دادیم.

- **استفاده از fairseq_generate برای ترجمه دادگان ارزیابی**

به کمک دستور fairseq-generate می‌توانید دقت مدل بر روی داده‌های تست را محاسبه کنید. خروجی این دستور را در new_eval.txt ذخیره می کنیم. در این فایل هر ۵ خط شامل یک شماره‌ی یکسان هستند که نشان دهنده‌ی این است که به همدیگر مربوط اند. حرف S نشان دهنده‌ی جمله در زبان مبدا است. هرف T نشان دهنده‌ی جمله در زبان مقصد است. حرف H نشان دهنده جمله‌ی تولیدی مدل است. حرف D نشان دهنده‌ی detokenize شده‌ی جمله‌ی تولید شده توسط مدل است. در نهایت P هم نشان دهنده‌ی امتیاز هر توکن در خروجی است. در انتهای فایل تولید شده توسط این دستور، امتیاز BLEU برای داده‌های تست محاسبه شده است.

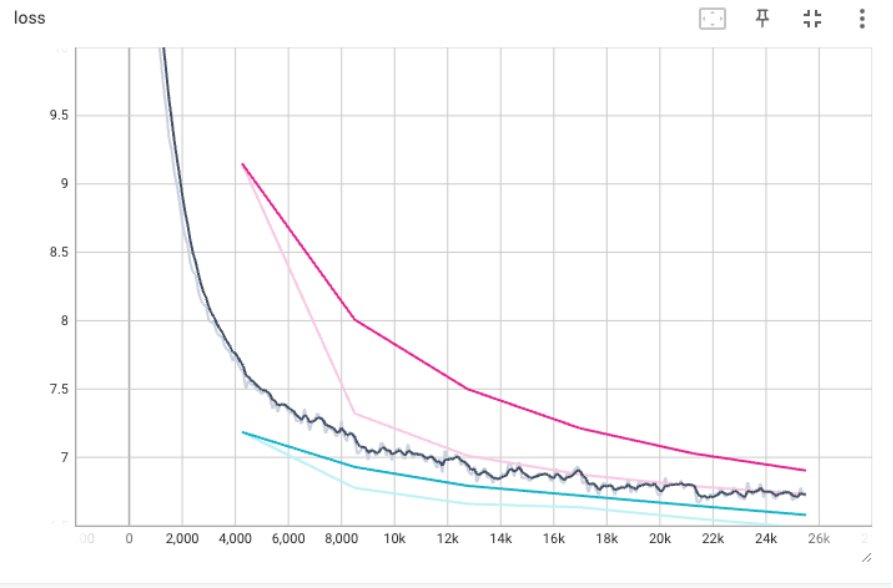
- **استفاده از TensorBoardبرای بررسی فرآیند آموزش**

در این مرحله با استفاده از tensorBoard می خواهیم نمودار های loss، BLEU و best Bleu را رسم کنیم. همچنین با استفاده از ابزار tensorboard مقدار loss را نیز به صورت csv در خروجی دریافت می کنیم.

فایل log مربوط به نمودار loss با فرمت **csv** برای این مرحله در پوشه /log/q1_part1 ذخیره شده است

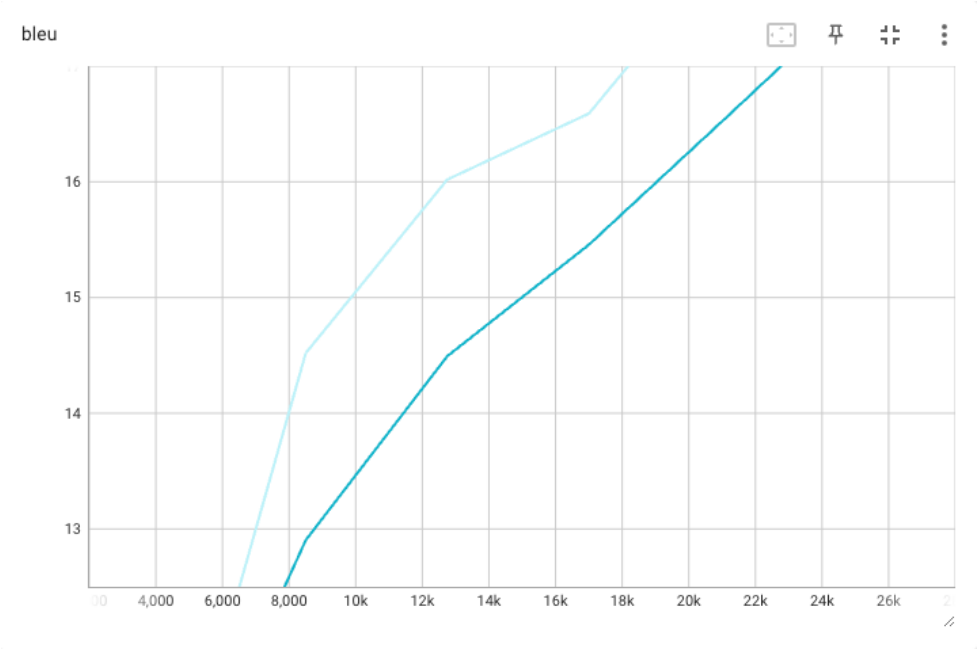
در این حالت مقدار BLEU برای دادگان ارزیابی(خط آخر فایل خروجی fairseq_generate) برابر با ۲۴,۷۳ می باشد.

نمودار loss : این نمودار شامل سه داده train،inner_train وvalidation می باشد.
(صورتی:train – فیروزه ای: valid -- خاکستری :train_inner)



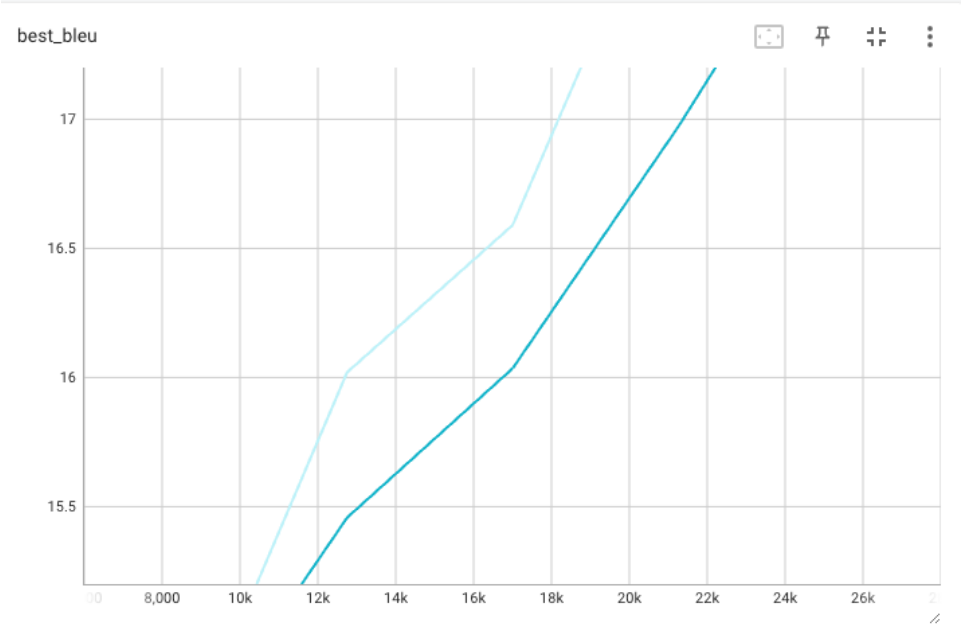
شکل ۱ : نمودار loss برای سوال یک بدون استفاده از BPE

نمودار bleu : این نمودار بر روی داده های validation رسم شده است.



شکل ۲ : نمودار BLEU برای سوال یک بدون استفاده از BPE

نمودار best_bleu : این نمودار بر روی داده های validation رسم شده است.



شکل ۳ : نمودار best BLEU برای سوال یک بدون استفاده از BPE

در انتها نیز فایل های لاگ دانلود شده است. دلیل این کار آن بود که به دلیل خطا در دانلود فایل csv از کولب، ما tensorboard را به صورت local نیز ران کردیم تا فایل های csv را خروجی بگیریم.

○ بخش دوم : مدل با استفاده از BPE

حال در این مرحله می خواهیم مدل را با استفاده از دادگان پیش پردازش شده با استفاده از BPE بسازیم.

○ آموزش مدل BPE برای دادگان انگلیسی و فارسی

در این قسمت با استفاده از کتابخانه sentencePiece و دستور SentencePieceTrainer مجموعه دادگان آموزش فارسی و انگلیسی را به صورت مجزا آموزش می دهیم. همانطور که در صورت سوال گفته شده در این قسمت مقدار vocab را برابر با 5k در نظر می گیریم.

- **پردازش دادگان با مدل های BPE**

حال برای پردازش دادگان، تابع bpe_tokenizer را تعریف می کنیم. این تابع مدل آموزش داده شده در قسمت قبل و فایل ورودی و آدرس ذخیره فایل خروجی را به عنوان ورودی دریافت می کند. سپس به کمک مدل آموزش داده شده دادگان train، validation و test را بر روی هر دو زبان با BPE توکنایز می کند و درپوشه data_tokenized_bpe ذخیره می کند.

- **پیش پردازش دادگان با استفاده از fariseq-preprocess**

همانطور که در hands-on گفته شده برای استفاده از fair-seq ابتدا باید از دستور fairseq-preprocess استفاده کرد. این دستور دیکشنری های fairseq را تولید می کند و داده ها را binarize می کند. در این دستور داده های آموزش، اعتبارسنجی و تست به عنوان ورودی داده می شود و مسیر مشخص شده data_bin مسیر ذخیره خروجی می باشد همچنین با استفاده از دستورات nwordssrc و nwordstgt اندازه vocab را برابر با ۴۰۰۰۰ تعیین می کنیم.

- **آموزش مدل با استفاده از fairseq-train**

پس از ذخیره داده های binarize شده، حال می خواهیم مدل را آموزش دهیم. در این قسمت ابتدا خروجی قسمت قبل را که در محل data_bin ذخیره کردیم به عنوان ورودی اول می دهیم. سپس در قسمت arch-- معماری مدل را برابر با LSTM با مکانیزم attention قرار می دهیم. سپس بهینه ساز را برابر با adam قرار داده و پارامتر های beta1=0.9 و beta2=0.98 قرار می دهیم .همینطور نرخ یادگیری را با استفاده از lr 25e-4-- مقدار دهی اولیه می کنیم و با استفاده از lr-scheduler inverse_sqrt-- آن را تطبیقی می کنیم. در ادامه مقدار dropout را برابر با ۰.۲۵ قرار می دهیم. سپس تابع هزینه را برابر با label smoothed cross entropy قرار داده و مقدار label smoothing را برابر با 0.2 قرار می دهیم. همچنین تعداد ایپاک را برابر با ۶ قرار دادیم.

- **استفاده از fairseq_generate برای ترجمه دادگان ارزیابی**

به کمک دستور fairseq-generate می‌توانید دقت مدل بر روی داده‌های تست را محاسبه کنید. خروجی این دستور را در new_eval.txt ذخیره می کنیم. در این فایل هر ۵ خط شامل یک شماره‌ی یکسان هستند که نشان دهنده‌ی این است که به همدیگر مربوط اند. حرف S نشان

دهنده‌ی جمله در زبان مبدا است. هرف T نشان دهنده‌ی جمله در زبان مقصد است. حرف H نشان دهنده جمله‌ی تولیدی مدل است. حرف D نشان دهنده‌ی detokenize شده‌ی جمله‌ی تولید شده توسط مدل است. در نهایت P هم نشان دهنده‌ی امتیاز هر توکن در خروجی است. در انتهای فایل تولید شده توسط این دستور، امتیاز BLEU برای داده‌های تست محاسبه شده است.

- **استفاده از TensorBoardبرای بررسی فرآیند آموزش**

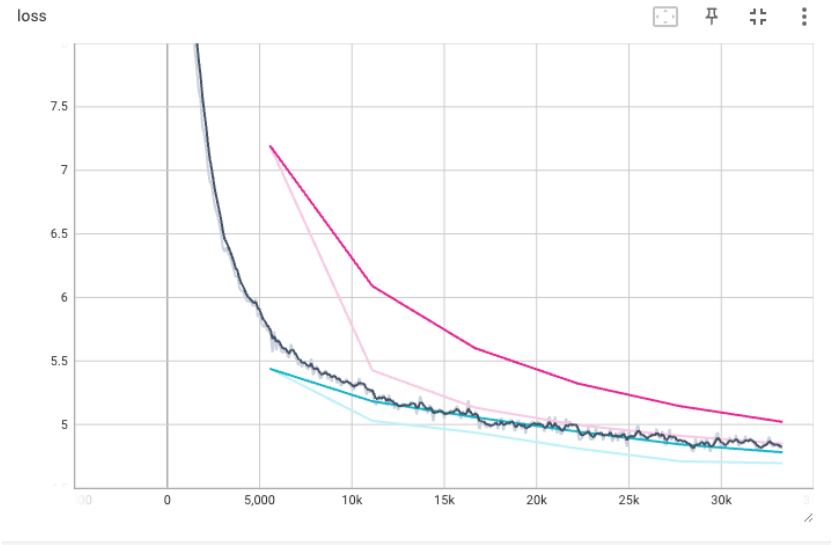
در این مرحله با استفاده از tensorBoard می خواهیم نمودار های loss، BLEU و best Bleu را رسم کنیم. همچنین با استفاده از ابزار tensorboard مقدار loss را نیز به صورت csv در خروجی دریافت می کنیم.

فایل log مربوط به نمودار loss با فرمت csv برای این مرحله در پوشه /log/q1_part2_bpe ذخیره شده است

در این حالت مقدار BLEU برای دادگان ارزیابی(خط آخر فایل خروجی fairseq_generate) برابر با ۲۶,۴۵می باشد.

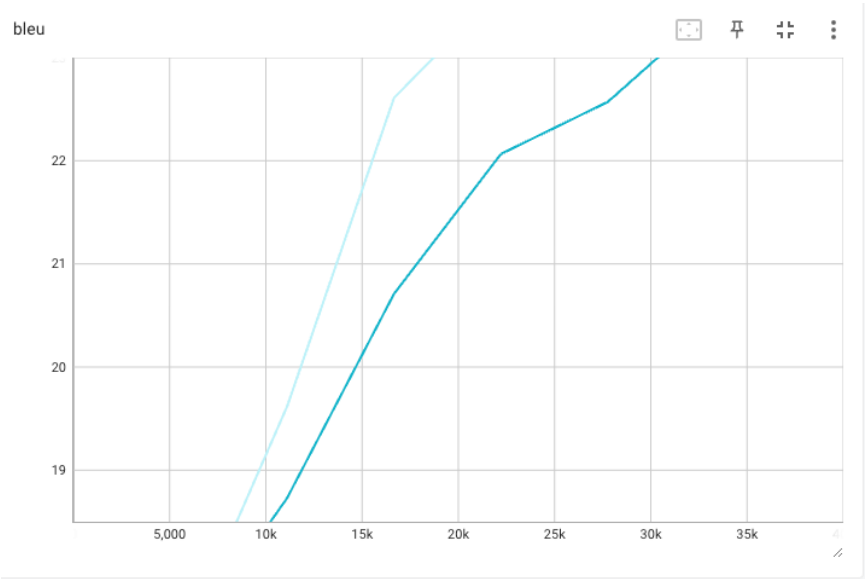
مقدار bleu برای حالت با توکنایز bpe بالاتر از قسمت بدون توکنایز است که نشان از بهبود مدل در این حالت دارد که همان نتیجه ای بود که انتظار داشتیم. زیرا در این قسمت یک مرحله توکنایزر اضافه کردیم.

نمودار loss : این نمودار شامل سه داده train،inner_train وvalidation می باشد.
(صورتی:train – فیروزه ای: valid -- خاکستری :train_inner)



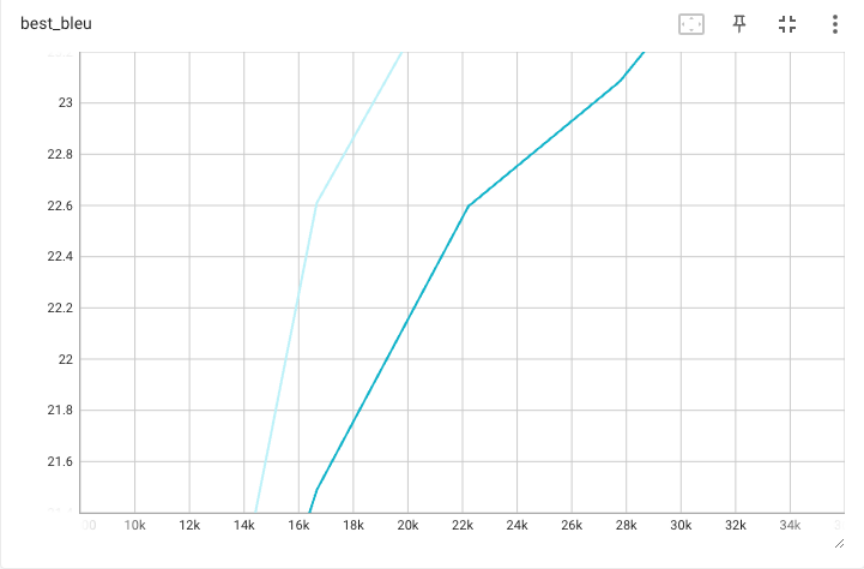
شکل ۴ : نمودار loss برای سوال یک با استفاده از BPE

نمودار bleu : این نمودار بر روی داده های validation رسم شده است.



شکل ۵ : نمودار BLEU برای سوال یک با استفاده از BPE

نمودار best_bleu : این نمودار بر روی داده های validation رسم شده است.



شکل 6 : نمودار best BLEU برای سوال یک با استفاده از BPE

سوال دوم

در این سوال می خواهیم از مدل mBERT که یک مدل از پیش آموزش داده شده است استفاده کنیم. همچنین در این سوال می خواهیم با استخراج وزن های embedding مدل گفته شده، به عنوان مقدار اولیه وزن های شبکه خود استفاده کنیم.

بخش اول : وزن ها فریز شده باشد

در ابتدا کتابخانه های مورد نیاز را نصب کرده و فرا میخوانیم. سپس ابتدا دادگان را در کولب load کرده و سپس مراحل زیر را به ترتیب دنبال می کنیم:

دانلود Bert Tokenizer و Bert Model

در این قسمت tokenizer و model را مطابق آنچه در صورت سوال گفته شده تعریف می کنیم که مدل از پیش آموزش داده شده می باشد(mBERT). پیاده سازی این مرحله به شرح زیر است:

```
tokenizer = BertTokenizer.from_pretrained('bert-base-multilingual-cased')
model = BertModel.from_pretrained("bert-base-multilingual-cased")
```

پردازش دادگان AFEC با استفاده از Bert Tokenizer

در این قسمت تابع tokenize_data را تعریف می کنیم. که آدرس ورودی دادگان فارسی و انگلیسی را گرفته، آن ها را با استفاده از توکنایزر تعریف شده توکنایز کرده و سپس در آدرس خروجی ذخیره می کند. این تابع را سه بار و بر روی داده های آموزش، validation و تست اجرا می کنیم و داده های توکنایز شده را در پوشه data_tokenized ذخیره می کنیم.

```
tokenize_data(input_file_en, input_file_fa, output_file_en, output_file_fa)
```

پیش پردازش دادگان با استفاده از fairseq-preprocess

همانطور که در hands-on گفته شده برای استفاده از fair-seq ابتدا باید از دستور fairseq-preprocess استفاده کرد. این دستور دیکشنری های fairseq را تولید می کند و داده ها را binarize می کند. در این دستور داده های آموزش، اعتبارسنجی و تست به عنوان ورودی داده می شود و مسیر مشخص شده data_bin_q2_freeze مسیر ذخیره

خروجی می باشد همچنین با توجه به اینکه tokenizerبرای هر دو زبان مبدا و مقصد یکسان است، دستور joined-dictionary را اضافه می کنیم.

```
    --joined-dictionary
```

- **ذخیره وزن های لایه embedding شبکه Bert با فرمت txt**

در این مرحله قصد داریم وزن های لایه embedding را در یک فایل با فرمت مناسب ذخیره کنیم. فرمت این فایل در صورت سوال آمده است و خط اول باید vocab_size و embedding_size بوده و خط های بعدی شامل یکی از اعضای vocab و بردار embedding مربوط به آن می باشد. در این مرحله با توجه به آنکه vocab_size بسیار بزرگ بود و باعث کرش کردن محیط کولب می شد و این مدل یک مدل چند زبانه بود، vocab_size را برابر با ۵۰۰۰ قرار دادیم. در نهایت این وزن ها را در فایل embedding_weights.txt ذخیره کردیم.

- **آموزش مدل با استفاده از fairseq-train**

پس از ذخیره داده های binarize شده در قسمت preprocess، حال می خواهیم مدل را آموزش دهیم. در این قسمت ابتدا خروجی قسمت قبل را که در محل data_bin_q۲ ذخیره کردیم به عنوان ورودی اول می دهیم. سپس وزن های embeddingذخیره شده در مرحله قبل را به عنوان وزن های اولیه انکودر و دیکودر خواهیم داد. همچنین سایز embedding ها را برای مدل تعیین کرده و برابر با ۷۶۸ قرار می دهیم. همچنین در این قسمت چون وزن های ما فریز شده است از دستور های فریز کردن وزن نیز استفاده کردیم.

```
    --encoder-embed-path "/content/embedding_weights.txt" \
    --decoder-embed-path "/content/embedding_weights.txt" \
    --decoder-embed-dim 768 \
    --decoder-out-embed-dim 768\
    --encoder-embed-dim 768 \
    --share-all-embeddings \
    --encoder-freeze-embed \
    --decoder-freeze-embed \
```

سپس در قسمت arch-- معماری مدل را برابر با LSTM با مکانیزم attention قرار می دهیم. سپس بهینه ساز را برابر با adam قرار داده و پارامتر های beta1=0.9 و beta2=0.98 قرار می دهیم .همینطور نرخ یادگیری را با استفاده از lr 25e-4-- مقدار

دهی اولیه می کنیم و با استفاده از --lr-scheduler inverse_sqrt آن را تطبیقی می کنیم. در ادامه مقدار dropout را برابر با ۰.۲۵ قرار می دهیم. سپس تابع هزینه را برابر با label smoothed cross entropy قرار داده و مقدار label smoothing را برابر با 0.2 قرار می دهیم.

- **استفاده از fairseq_generate برای ترجمه دادگان ارزیابی**

به کمک دستور fairseq-generate می‌توانید دقت مدل بر روی داده‌های تست را محاسبه کنید. خروجی این دستور را در فایل new_eval.txt ذخیره می کنیم. در این فایل هر ۵ خط شامل یک شماره‌ی یکسان هستند که نشان دهنده‌ی این است که به همدیگر مربوط اند. حرف S نشان دهنده‌ی جمله در زبان مبدا است. هرف T نشان دهنده‌ی جمله در زبان مقصد است. حرف H نشان دهنده جمله‌ی تولیدی مدل است. حرف D نشان دهنده‌ی detokenize شده‌ی جمله‌ی تولید شده توسط مدل است. در نهایت P هم نشان دهنده‌ی امتیاز هر توکن در خروجی است. در انتهای فایل تولید شده توسط این دستور، امتیاز BLEU برای داده‌های تست محاسبه شده است.

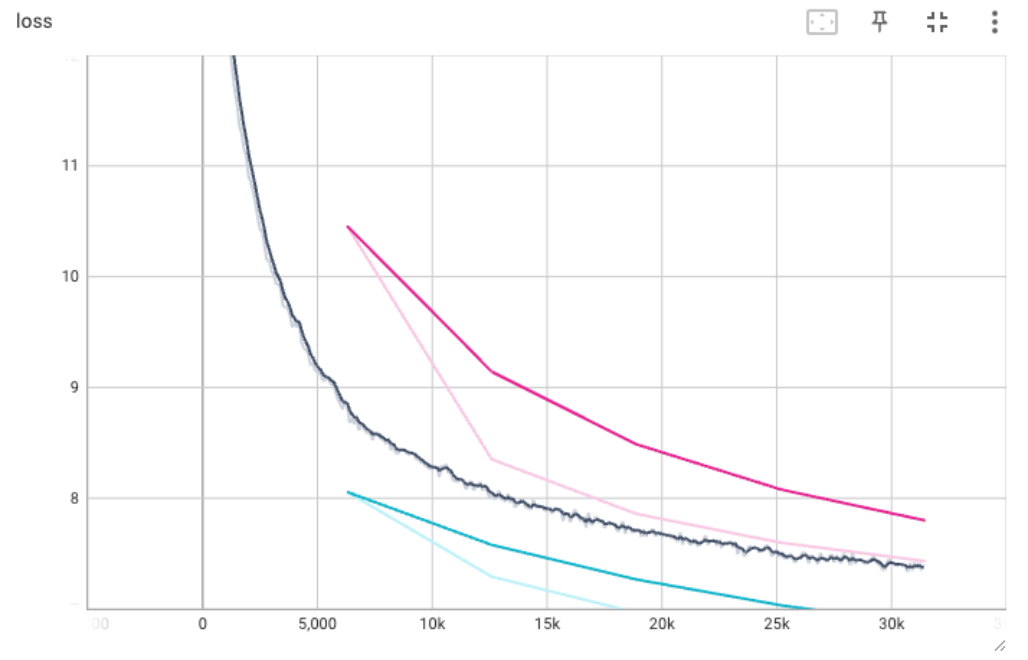
- **استفاده از TensorBoardبرای بررسی فرآیند آموزش**

در این مرحله با استفاده از tensorBoard می خواهیم نمودار های loss، BLEU و best Bleu را رسم کنیم. همچنین با استفاده از ابزار tensorboard مقدار loss را نیز به صورت csv در خروجی دریافت می کنیم.

فایل log مربوط به نمودار loss با فرمت csv برای این مرحله در پوشه /log/q2_part1_freeze ذخیره شده است

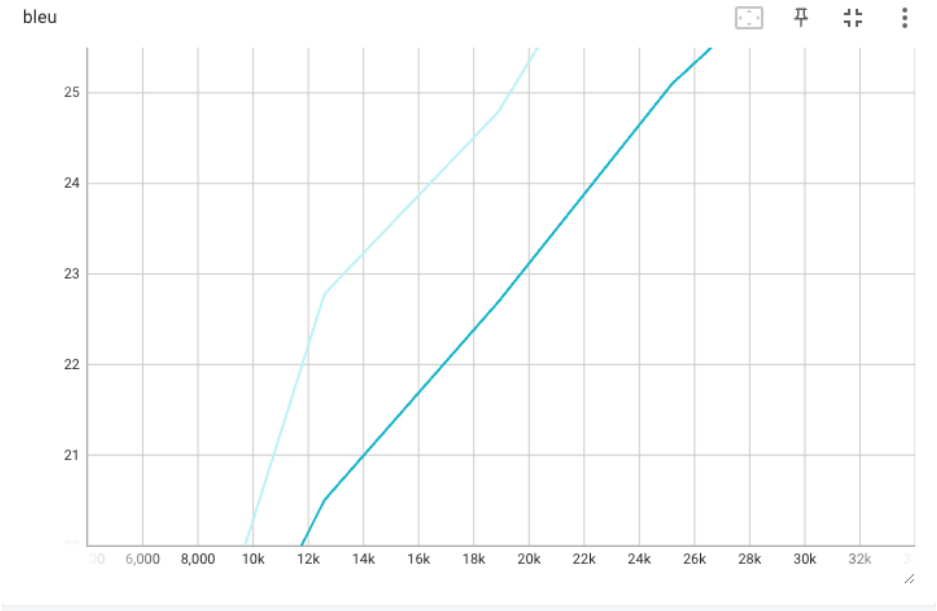
در این حالت مقدار BLEU برای دادگان ارزیابی(خط آخر فایل خروجی fairseq_generate) برابر با۱۸,۶۲می باشد.

نمودار loss : این نمودار شامل سه داده train،inner_train و validation می باشد.
(صورتی:train – فیروزه ای: valid -- خاکستری :train_inner)



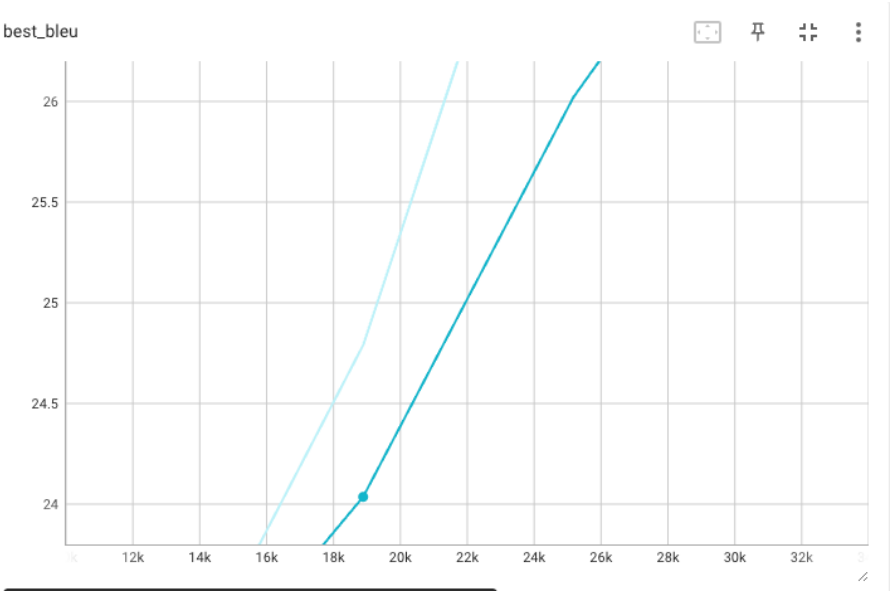
شکل 7 : نمودار loss برای سوال دو با وزن فریز شده

نمودار BLEU : این نمودار بر روی داده های validation رسم شده است.



شکل ۸ : نمودار BLEU برای سوال دو با وزن فریز شده

نمودار best BLEU : : این نمودار بر روی داده های validation رسم شده است.



شکل 9 : نمودار best BLEU برای سوال دو با وزن فریز شده

- **بخش دوم : وزن ها فریز نشده باشد:**

این حالت همانند قسمت قبل است فقط در مرحله train دو خط مربوط به freeze کردن وزن ها را پاک می کنیم. بقیه موارد مانند قسمت قبل می باشد.

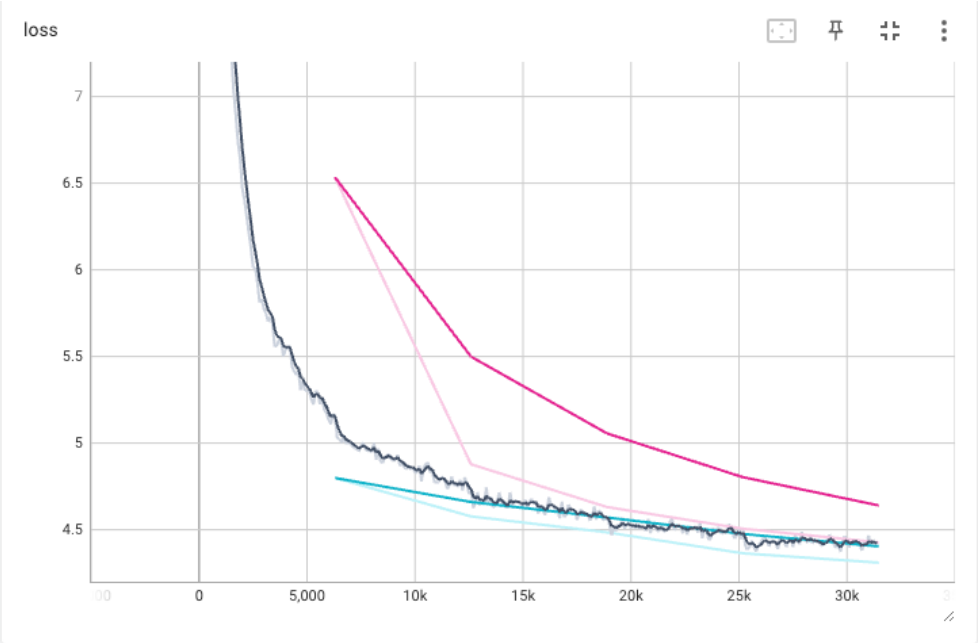
در این مرحله با استفاده از tensorBoard می خواهیم نمودار های loss، BLEU و best Bleu را رسم کنیم. همچنین با استفاده از ابزار tensorboard مقدار loss را نیز به صورت csv در خروجی دریافت می کنیم.

فایل log مربوط به نمودار loss با فرمت csv برای این مرحله در پوشه /log/q2_part2_Nofreeze ذخیره شده است

در این حالت مقدار BLEU برای دادگان ارزیابی(خط آخر فایل خروجی fairseq_generate) برابر با ۳۶,۱۵می باشد.

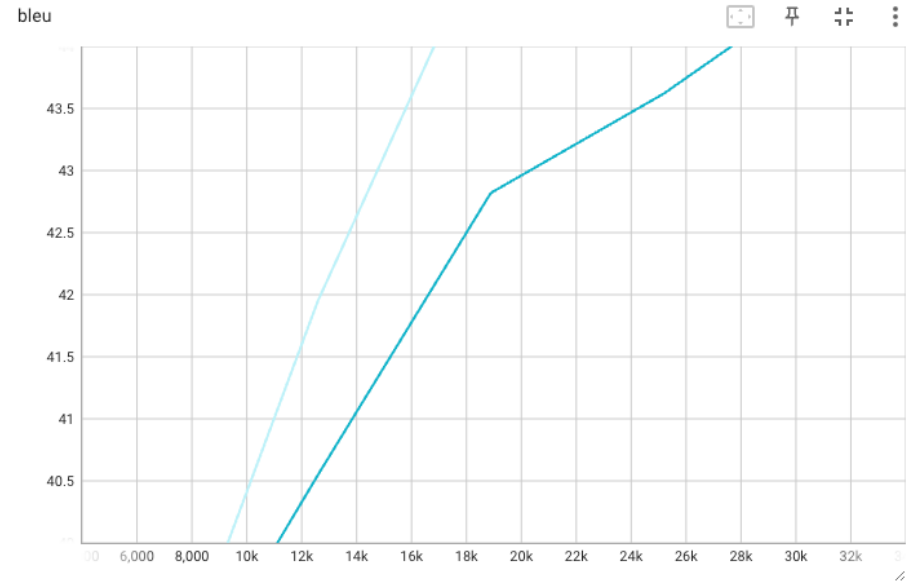
فریز نکردن وزن ها در قسمت دوم باعث افزایش دو برابری مقدار bleu نسبت به مرحله قبل نشان می دهد پس فریز نکردن وزن ها باعث بهبود مدل شده است.

نمودار loss : این نمودار شامل سه داده train،inner_train و validation می باشد.
(صورتی:train – فیروزه ای: valid -- خاکستری :train_inner)



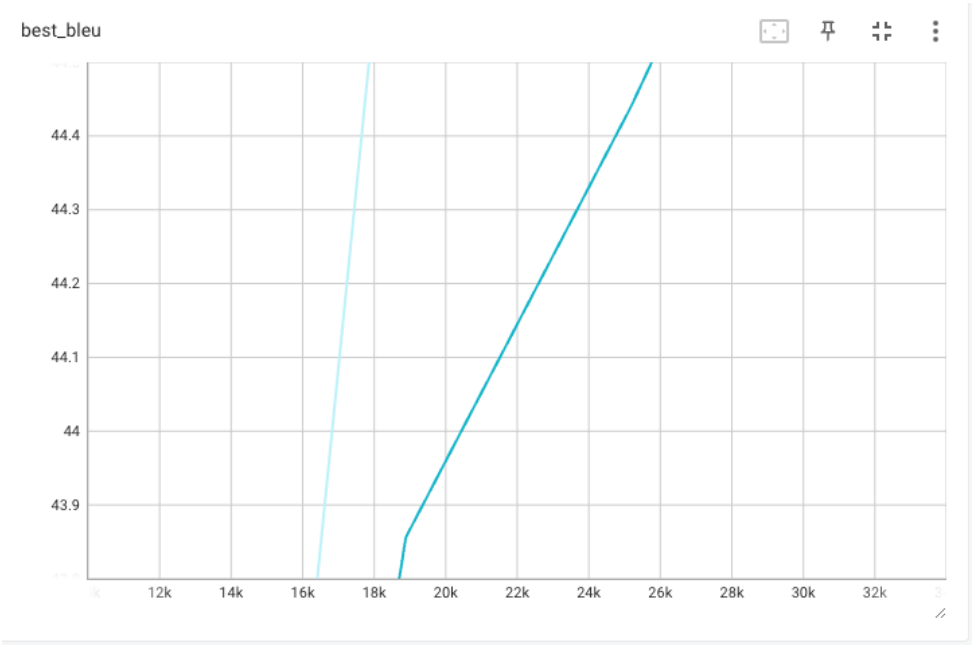
شکل ۱۰ : نمودار loss برای سوال دو با وزن فریز نشده

نمودار BLEU : این نمودار بر روی داده های validation رسم شده است.



شکل ۱۱ : نمودار BLEU برای سوال دو با وزن فریز نشده

نمودار best BLEU : : این نمودار بر روی داده های validation رسم شده است.



شکل 12 : نمودار best BLEU برای سوال دو با وزن فریز نشده