بنام خدا

دانشکده‌ی مهندسی برق و کامپیوتر

درس پردازش زبان‌های طبیعی

شایان واصف احمدزاده , 810197603

امیرحسین دبیری اقدم , 810197502

تمرین کامپیوتری **5**

Machine Translation

روز آپلود : 18 خرداد

اساتید : دکتر فِیلی و دکتر یعقوب زاده

Table of Contents

Preprocessing[1]

Essential ones

.1

همانطور که اشاره شد، اولین و اصلی ترین پیش پردازش دادگان مربوط به اعمال BPE (برای Tokenization) بر روی دادگان آموزش فارسی و انگلیسی می‌باشد. برای اعمال BPE نیاز است تا Pre-tokenization انجام شود (ابتدا باید کلمات با فاصله از هم جدا شده باشند) که دیتاست داده شده خود اینچنین است. همچنین در ادامه دو روش دیگر پیش‌پردازش را شرح می‌دهیم تا بتواند به آموزش مدل کمک کند. بسته به اینکه دادگان فارسی یا انگلیسی باشند، پردازش متفاوتی انجام میدهیم :

- دادگان فارسی: برای دادگان فارسی ابتدا تمامی نیم فاصله‌ها را به فاصله تبدیل می‌کنیم، برای این کار لازم است تا به جای کد اسکی '\u200c' فاصله قرار دهیم تا در فرآیند Tokenization به BPE کمک کند. همچنین برای consistency، تمامی single quotation(') ها را با double quotation(") تبدیل کردیم از آنجا که تفاوتی با یکدیگر ندارند.
- دادگان انگلیسی: برای دادگان انگلیسی مانند دادگان فارسی تمامی single quotation ها را با double quotation تبدیل کردیم و همچنین تمامی لغات را از upper case به lower case تغییر دادیم.

.2

به صورت کلی Tokenization (اعمال BPE) اساسی ترین پیش‌پردازش برای هر دو زبان فارسی و انگلیسی به شمار می‌رود. همچنین در دادگان فارسی مهم‌ترین پیش‌پردازش، مربوط به حذف یا نگه داری نیم فاصله می‌باشد زیرا تاثیر محسوسی در فرآیند Tokenization می‌گذارد و طبیعتا در عملکرد مدل تاثیرگذار هست.

در دادگان انگلیسی مهم‌ترین پیش‌پردازش، مربوط به upper یا lower case کردن لغات می‌باشد، زیرا خیلی اوقات اسم‌های معروف یا احساسات هیجانی به صورت upper case نوشته می‌شوند. ولی در Task بخصوصی مثل ترجمه این موضوع اهمیت چندانی ندارد بنابراین تمام کلمات را lower case کرده‌ایم تا با این کار به Tokenization کلماتی که صرفاً از نظر case متفاوت هستند اما معمولاً از نظر معنایی متفاوت نیستند، کمک کند .

Implementation

در ابتدا دو پیش پردازش ذکر شده را بر روی دادگان آموزش و تست و ارزیابی اعمال می‌کنیم :

```
src = ['train.fa','train.en','test.fa','test.en','valid.fa','valid.en']
```

[1] نتبوک کدها و نیز به طور کلی تمام فایل‌های مربوط به این پروژه در اینجا قابل مشاهده است.

```
Dst =
['train_prc.fa','train_prc.en','test_prc.fa','test_prc.en','valid_prc.fa','valid_
prc.en']

for i,file in enumerate(src):

  with open('/content/drive/MyDrive/Notebooks/NLP/CA5/data/'+ file,'r') as f1:

      lines = f1.read()

  if file[-2:]=='fa':

    lines = lines.replace('\u200c',' ')

    lines = lines.replace("'", '"')

  else :

    lines = lines.lower()

    lines = lines.replace("'", '"')

  f1.close()

  with open('/content/drive/MyDrive/Notebooks/NLP/CA5/data/'+ Dst[i],'w') as f2:

    f2.write(lines)

  f2.close()
```

در ادامه به کمک کتابخانه subword-nmt، پیش پردازش BPE را روی دادگان آموزش فارسی و انگلیسی اعمال می‌کنیم. به کمک دستور learn-bpe مرحله آموزش BPE را انجام می‌دهیم .تعداد (حداکثر) Token را برای زبان انگلیسی برابر 25000 و برای زبان فارسی برابر 20000 در نظر می‌گیریم :

```
subword-nmt learn-bpe -s 25000 <drive/MyDrive/Notebooks/NLP/CA5/data/train_prc.en
> drive/MyDrive/Notebooks/NLP/CA5/data/code.en

100% 25000/25000 [00:38<00:00, 652.44it/s]

subword-nmt learn-bpe -s 20000 <
drive/MyDrive/Notebooks/NLP/CA5/data/train_prc.fa >
drive/MyDrive/Notebooks/NLP/CA5/data/code.fa

100% 20000/20000 [00:25<00:00, 772.08it/s]
```

در ادامه الگوریتم BPE آموزش داده شده را بر روی دادگان آموزش، تست و ارزیابی به کمک دستور apply-bpe اعمال می‌کنیم :

English Data:

```
subword-nmt apply-bpe -c drive/MyDrive/Notebooks/NLP/CA5/data/code.en <
drive/MyDrive/Notebooks/NLP/CA5/data/train_prc.en >
drive/MyDrive/Notebooks/NLP/CA5/data/train-bpe.en

subword-nmt apply-bpe -c drive/MyDrive/Notebooks/NLP/CA5/data/code.en <
drive/MyDrive/Notebooks/NLP/CA5/data/valid_prc.en >
drive/MyDrive/Notebooks/NLP/CA5/data/valid-bpe.en

subword-nmt apply-bpe -c drive/MyDrive/Notebooks/NLP/CA5/data/code.en <
drive/MyDrive/Notebooks/NLP/CA5/data/test_prc.en >
drive/MyDrive/Notebooks/NLP/CA5/data/test-bpe.en
```

Persian Data:

```
subword-nmt apply-bpe -c drive/MyDrive/Notebooks/NLP/CA5/data/code.fa <
drive/MyDrive/Notebooks/NLP/CA5/data/train_prc.fa >
drive/MyDrive/Notebooks/NLP/CA5/data/train-bpe.fa

subword-nmt apply-bpe -c drive/MyDrive/Notebooks/NLP/CA5/data/code.fa <
drive/MyDrive/Notebooks/NLP/CA5/data/valid_prc.fa >
drive/MyDrive/Notebooks/NLP/CA5/data/valid-bpe.fa

subword-nmt apply-bpe -c drive/MyDrive/Notebooks/NLP/CA5/data/code.fa <
drive/MyDrive/Notebooks/NLP/CA5/data/test_prc.fa >
drive/MyDrive/Notebooks/NLP/CA5/data/test-bpe.fa
```

در نهایت باید پیش پردازش بدست آمده را در قالب command line موجود در ابزار fairseq پیاده سازی کنیم :

```
fairseq-preprocess --tokenizer space --bpe subword_nmt -s en -t fa \
  --trainpref drive/MyDrive/Notebooks/NLP/CA5/data/train-bpe \
  --validpref drive/MyDrive/Notebooks/NLP/CA5/data/valid-bpe \
  --testpref drive/MyDrive/Notebooks/NLP/CA5/data/test-bpe \
  --destdir drive/MyDrive/Notebooks/NLP/CA5/data/summary --workers 20
```

همچنین command line مربوط در ابزار openNMT به صورت زیر می‌باشد :

```
onmt_build_vocab -config en-fa.yaml -n_sample -1
```

*فایل‌های ساخته شده بعد از پیش‌پردازش در پوشه تمرین کامپیوتری قرار داده شده است.

Model Selection

ما در این پروژه مدل ترجمه ماشینی مبتنی بر Transformer با ساختار مشابه مقاله Attention is all you need[2] را با استفاده از دو ابزار FairSeq[3] و OpenNMT[4] پیاده سازی کردیم که در ادامه به توضیح آنها می‌پردازیم.

FairSeq[5]

Important Hyperparameters

در ادامه به 10 تا از مهم ترین پارامترهای ابزار Fairseq می‌پردازیم و کارکرد هر یک را مختصر شرح می‌دهیم :

1. Model Configuration: یکی از پارامتر های مهم ، تعیین معماری مدل می‌باشد که ما بر روی 'Transformer' قرار می‌دهیم. نحوه تعیین این پارامتر در command line بصورت زیر می‌باشد :

```
--arch transformer
```

2. Loss Function : یکی از پارامترهای مهم دیگر تعیین Loss Function می‌باشد. طبق تنظیمات در مقاله، از تابع Cross Entropy به همراه Lable smoothing استفاده می‌شود. همچنین مقدار Weight decay را برابر 0 در نظر می‌گیریم:

```
--criterion label_smoothed_cross_entropy --label-smoothing 0.1 --weight-decay 0.0
```

3. Optimizer :پارامتر مهم بعدی، تعیین نوع تابع بهینه ساز می‌باشد که طبق توضیحات مقاله، از بهینه ساز Adam با پارامترهای beta زیر استفاده می‌شود :

```
--optimizer adam --adam-betas '(0.9, 0.98)'
```

4. Batch Size :پارامتر بعدی، تعداد نمونه‌های درون هر دسته (Batch) از دادگان می‌باشد که برای اینکه دیتاست داده شده نسبت به دیتاست مقاله کوچکتر می‌باشد و همچنین به دلیل مشکلات ناشی از اشغال حافظه GPU، این پارامتر را برابر هر دوی دادگان آموزش و ارزیابی برابر 64 قرار دادیم:

```
--batch-size 64
```

```
--batch-size-valid 64
```

5. Encoder/Decoder Embedding/ffn Embedding size: پارامتر مهم بعدی، تعیین سایز Embedding های مربوط به لایه Encoder و Decoder می‌باشد که طبیعتا رابطه مستقیمی با تعداد پارامترهای مدل نیز دارد. سایز لایه

[2] https://arxiv.org/pdf/1706.03762.pdf
[3] https://github.com/facebookresearch/fairseq
[4] https://github.com/OpenNMT/OpenNMT-py

[5] نتبوک مربوط به این ابزار در کنار این گزارش پیوست شده است.

Embedding کلمات ورودی را برابر 512 و Embedding شبکه FeedForward خروجی را برابر 2048 قرار می‌دهیم.

```
--encoder-embed-dim 512 --decoder-embed-dim 512
```

```
--encoder-ffn-embed-dim 2048 --decoder-ffn-embed-dim 2048
```

6. Encoder/Decoder number of layers: پارامتر بعدی تعداد لایه‌های Transformer در دو بخش Enoder و Decoder می‌باشد که مجددا رابطه مستقیمی با تعداد پارامتر مدل دارد. مقدار این پارامتر برای هر دو مورد را برابر 6 (small size) قرار می‌دهیم.

```
--encoder-layers 6 --decoder-layers 6
```

7. Drop Out: پارامتر مهم دیگر، تعیین احتمال dropout می‌باشد. خود این پارامتر در دو جا مختلف Transformer ها تعریف می‌شود: بعد از توابع فعال ساز که به آن dropout گفته می‌شود و در هنگام attention که به آن dropout attention می‌گوییم. این پارامتر از این جهت که مدل در حین آموزش، بیش از حد جملات مرجع را حفظ نکند و به اصطلاح overfit نشود، مهم است. مقدار هر دوی آنها را برابر 0.1 در نظر می‌گیریم :

```
--dropout 0.1 --attention-dropout 0.1
```

8. Source/Target Language: طبیعتا یکی از ابتدایی ترین پارامترهای مدل، تعیین زبان مبدا و مقصد می‌باشد که زبان مبدا انگلیسی و مقصد فارسی می‌باشد. این پارامتر برای انتخاب درست فایل‌های آموزش و ارزیابی مربوط به زبان مبدا و مقصد به ابزار داده می‌شود.

```
--source-lang en --target-lang fa
```

9. Learning Rate: تعیین learning rate بسیار حائز اهمیت است از آن جهت که مدل در epoch های متوالی بتواند یادگیری مناسبی داشته باشد. طبیعتاً انتظار داریم که هر چه در آموزش یک مدل جلو می‌رویم، به نقاط مینمم (محلی) نزدیکتر شویم و بنابراین اگر مقدار learning rate را بتوان با تعداد epoch طی شده تنظیم کرد، می‌توان عملکرد بهتری از مدل را شاهد بود که به آن learning rate scheduler می‌گویند که می‌تواند روی حالات مختلفی تنظیم شود که در سرعت تغییرات learning rate موثر است. طبق توضیحات مقاله، از حالت inverse squared بهره می‌گیریم. همچنین پارامتر دیگری بنام warm up نیز قابل تنظیم است به طوریکه learning rate تا تعداد step های طی شده مدل تا مقدار warm up افزایش بیابد و سپس با یک نرخ معکوس مربعی کاهش پیدا می‌کند. در واقع در ابتدای کار که مدل تازه شروع به یادگیری کرده است، سرعت یادگیری مدل را افزایش می‌دهد و با نزدیک شدن مدل به نقطه بهینه سرعت یادگیری کاهش می‌یابد؛ ولی باید دقت کرد که تنظیم نامناسب این پارامتر می‌تواند باعث کندی مدل یا واگرایی آن نیز بشود.

رابطه مورد استفاده برای learning rate در مقاله مذکور بدین شکل می‌باشد:

$$lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$

طبق سعی و خطای انجام شده، مقدار warm up را برابر 1300 قرار دادیم که با مقداری که در مقاله تنظیم شده بود (4000) تفاوت دارد:

```
--lr 0.0007 --lr-scheduler inverse_sqrt --warmup-init-lr 1e-07 --warmup-updates 1300
```

10. Scoring: طبیعتاً برای بررسی عملکرد مدل، باید شیوه‌ای برای ارزیابی داشت. معیار متداول ارزیابی در ترجمه ماشین، معیار BLEU می‌باشد که ما هم از پیاده‌سازی آن توسط کتابخانه sacrebleu استفاده کردیم، البته که معیارهای loss،perplexity هم توسط مدل در فرآیند آموزش وجود داشت.(*تفاوت BLEU با sacrebleu در این مقاله[6] آمده است):

```
--eval-bleu --eval-bleu-args '{"beam": 5}'
--scoring sacrebleu
```

Fixed parameters

در ادامه به سه تا از پارامترهای مدل اشاره می‌کنیم که به دلیل محدودیت های موجود، امکان تغییر آنها وجود ندارد :

1. Encoder/Decoder number of layers: مدل‌های مرسوم Transformer دو نوع small و large دارند. مدل large که دارای 12 لایه در بخش‌های Encoder و Decoder می‌باشد، تعداد پارامترهای بسیار بیشتری نسبت به مدل small که 6 لایه دارد، می‌باشد. به دلیل محدودیت‌های سخت افزاری colab (حافظه، سرعت و...)، نمی‌توانیم از مدل با تعداد لایه‌های زیاد استفاده کنیم.
2. Encoder/Decoder Embedding size: مشابه قبل، به دلیل محدودیت موجود در colab، نمی‌توانیم از embedding های sparse استفاده کنیم. به همین دلیل از نمایش‌های Dense که سایز Embedding خیلی بزرگ نباشد (برای مثال 512) استفاده می‌کنیم.
3. Encoder/Decoder Feedforward Embedding Size: می‌دانیم که در انتهای هر لایه Transformer در بخش های Encoder و Decoder یک لایه FeedForward هم داریم. مشابه قبل، افزایش تعداد unit های این لایه باعث پیچدگی مدل و افزایش تعداد پارامترها می‌شود و به دلیل محدودیت حافظه colab این مقدار را برابر 2048 تنظیم می‌کنیم.

همچنین تعدادی پارامتر در ابزار fairseq وجود دارد که به دلیل محدودیت تعداد GPU در دسترس قابل تنظیم نیست :

1. Model Parallel Size: این پارامتر تعداد GPU های مورد استفاده برای پردازش موازی در حین آموزش مدل می‌باشد.
2. Device ID: این پارامتر مشخص می‌کند که کدام GPU در حین آموزش استفاده شود.

[6] https://arxiv.org/pdf/1804.08771.pdf

Sample Output per checkpoint

در ادامه سه جمله نمونه از دادگان تست انتخاب کردیم و ترجمه مرجع به همراه ترجمه مدل در هر check-point را آورده‌ایم :

Example 1: the Israel defense forces said that more than 20 mort@@ ars and rockets were subsequently fired into their territory.

Reference : نیروهای دفاعی اسراییل گفتند بیش از 20 خمپاره و موشک متعاقبا به قلمرو انان شلیک شد.

Hypothesis_check-point25 : اسرائیل گفت که نیروهای مسلح به آتش کشیده شده اند و در آن ها بیش از چهار تن دیگر زخمی شدند.

Hypothesis_check-point50 :اسرائیل گفت که نیروهای مسلح به آتش کشیده اند و بیش از 20 نفر نیروهای نظامی را کشته اند.

Hypothesis_check-point75 :اسرائیل گفت که نیروهای قدری بیش از 20 نفر مجروح شده اند و آتش به سوی آن آمده اند.

Hypothesis_check-point100 : اسرائیل گفت که نیروهای دفاعی را بیش از 20 نفر کشته و تن دیگر زخمی شدند.

Hypothesis_check-point125 : اسرائیل گفت که نیروهای دفاعی کشته شده اند و بیش از 20 تن به سوی خود اعتراف کرده اند.

Example 2: that "s how Washington dealt with the Soviet Union and China in the 1970 s and 80 s.

Reference : همین گونه بود که واشنگتن در دهه های 70 و 80 میلادی با جماهیر شوروی و چین کنار آمد.

Hypothesis_check-point25 : این توافق که چطور با اتحاد جماهیر شوروی در چین شد, و در دهه 70 است.

Hypothesis_check-point50 :این واشنگتن چطور است که در دهه های اتحاد جماهیر شوروی به شمار می رود.

Hypothesis_check-point75 :به همین دلیل است که واشنگتن با دهه های کشورهای زیادی در چین و دهه 70 میلادی ترک کردند.

Hypothesis_check-point100 :که واشنگتن چطور است نماینده اتحادیه اروپا با دهه های مهم و دهه 70 میلادی در آن زمان سفر شد.

Hypothesis_check-point125 : که واشنگتن ذکر شده اند و اتحادیه اروپا در دهه 70 میلادی در دهه 1970 متولد شد.

Example 3: officials killed two of the attackers, according to a regional police chief.

Reference :طبق گفته رئیس پلیس محلی, ماموران دو نفر از مهاجمین را کشته اند.

Hypothesis_check-point25 : به گزارش اکسپرس نیوز, مقامات پلیس گزارش داد که دو پلیس در این دو مامور پلیس را به قتل رسانده است .

Hypothesis_check-point50 : بنا بر این گزارش, مقامات افغان دو افسر پلیس عالی رتبه, یکی از افراد پلیس محلی در منطقه ای خود را کشتند .

Hypothesis_check-point75 : به گفته مقامات, دو افسر پلیس منطقه ای افغان را در جریان حمله کردند.

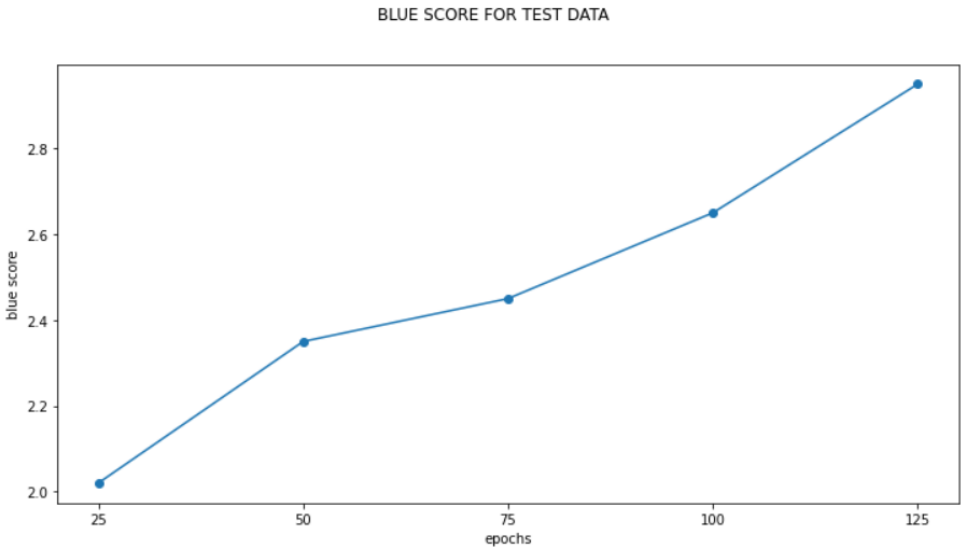
Hypothesis_check-point100 : به گفته مقامات, دو افسر پلیس منطقه ای در جریان پولی را کشتند.

Hypothesis_check-point125 : بنا بر اعلام مقامات این دو تن از مهاجمان به یک پاسگاه پلیس در منطقه ای پلیس محلی افغان کشته شده اند.

Scoring Results[7]

در ادامه، ارزیابی از مدل طی پنج checkpoint مختلف که هر checkpoint شامل 25 epoch از دادگان می‌باشد، ارائه می‌دهیم:

در اولین نمودار، معیار bleu را برحسب تعداد step برای دادگان تست نشان می‌دهد که تقریبا مشهود است که با افزایش step ها، معیار bleu رو به افزایش است که در نهایت به مقدار 2.95 می‌رسد.

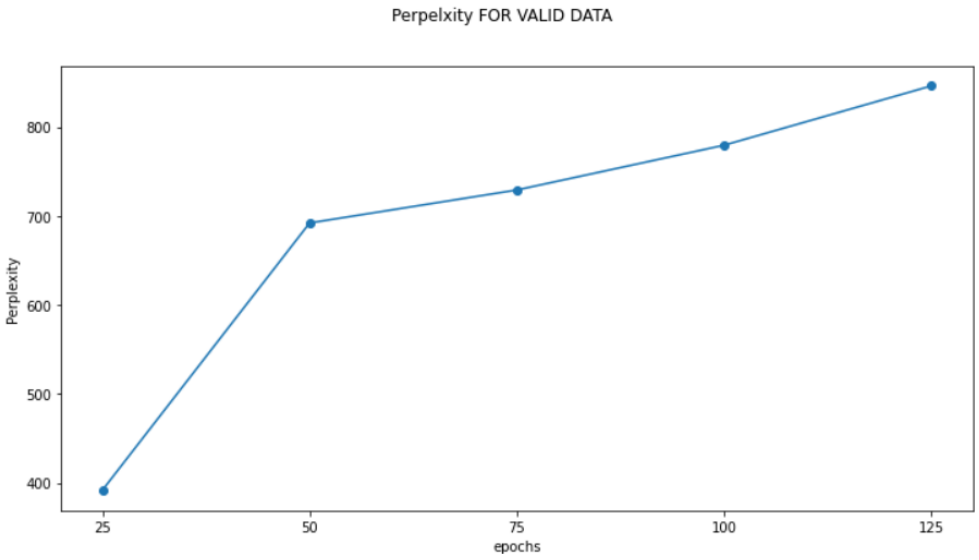


شکل 1 : نمودار امتیاز blue بر حسب Epoch برای دادگان تست

در نمودار دوم، معیار perplexity بر حسب تعداد step نمایش داده شده است. در 25 epoch اولیه از آموزش مدل، perplexity رو به کاهش بود که در شکل قابل دیدن نیست که نشان می‌دهد که جملات تولید شده از نظر زبانی درست هستند ولی با گذشت

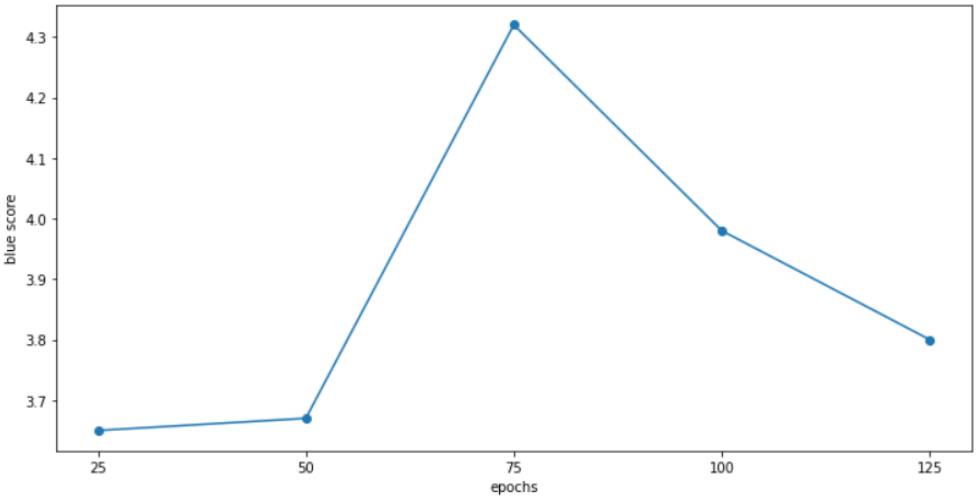
[7] خروجی مدل برای دیتاست تست در کنار این گزارش پیوست شده است.

زمان و افزایش معیار bleu، مقدار perplexity افزایش پیدا می‌کند به این معنی که خروجی ترجمه شده با مرجع تعداد unigram های مشترک زیادی دارند ولی شاید جمله بوجود آمده از نظر زبانی کمی مفهوم نیست.



شکل 2 : نمودار امتیاز perplexity بر حسب Epoch برای دادگان ارزیابی

در نهایت، نمودار bleu بر حسب تعداد epoch برای دادگان ارزیابی رسم شده است که مانند شکل 3، معیار bleu ابتدا افزایش یافته و سپس کاهش پیدا می‌کند. بیشترین مقدار bleu بدست آمده در 75 epoch می‌باشد که مقداری برابر 4.32 دارد.



شکل 3 : نمودار امتیاز blue بر حسب Epoch برای دادگان ارزیابی

OpenNMT[8]

Important Hyperparameters

در ادامه به 10 تا از مهم ترین پارامترهای ابزار OpenNMT می‌پردازیم؛ چون اکثراً نام و کارکرد مشابه با پارامترهای گفته شده برای ابزار FairSeq دارند تنها مواردی که متفاوت در نظر گرفته شده مقدار پارامتر یا ... را مختصراً شرح می‌دهیم. اکثر این پارامترها در فایل en-fa.yaml تنظیم شده و این فایل به عنوان ورودی ابزار در هنگام آموزش (و ترجمه) به صورت زیر داده می‌شود.

```
onmt_train -config en-fa.yaml --verbose
```

```
onmt_translate -model drive/MyDrive/CA5/checkpoints/onmt3/onmt_step_1800.pt -
src drive/MyDrive/CA5/data/test-bpe.en -tgt drive/MyDrive/CA5/data/test-
bpe.fa -output drive/MyDrive/CA5/data/pred2_1800.txt -gpu 0 -beam_size 5 -
batch_size 16384 -batch_type 'tokens' #-verbose
```

1. encoder_type و decoder_type (مشابه Model Configuration در FairSeq): یکی از پارامتر های مهم ، تعیین معماری انکدر و دیکدر مدل می‌باشد که ما بر روی 'Transformer' تنظیم می‌کنیم.
2. Loss Function : یکی از پارامترهای مهم دیگر تعیین تابع خطا یا همان Loss Function می‌باشد. طبق تنظیمات در مقاله، از تابع Cross Entropy به همراه Lable smoothing استفاده می‌شود.

```
label_smoothing:0.1
```

3. Optimizer: پارامتر مهم بعدی، تعیین نوع تابع بهینه ساز می‌باشد که طبق توضیحات مقاله، از بهینه ساز Adam با پارامتر های زیر استفاده می‌شود:

```
optim: "adam"
adam_beta2: 0.98
```

4. Batch Size: پارامتر بعدی، تعداد نوع و نمونه‌های درون هر دسته (Batch) از دادگان می‌باشد که بدین صورت قرار دادیم:

```
batch_type: "tokens"
normalization: "tokens"
max_grad_norm: 0
batch_size: 8192
valid_batch_size: 16384
```

[8] نتبوک مربوط به این ابزار در کنار این گزارش پیوست شده است.

5. Encoder/Decoder Embedding size: پارامتر مهم بعدی، تعیین سایز Embedding های مربوط به لایه FeedForward در بلوک‌های transformer می‌باشد که آن را برابر 2048 قرار می‌دهیم. همچنین اندازه بردار متناظر با هر کلمه را مشابه مقاله برابر با 512 می‌گذاریم. چون لازم است که positional embedding هم داشته باشیم در مدل‌های transformer، آن را هم فعال می‌کنیم:

```
transformer_ff: 2048

word_vec_size: 512

position_encoding: true
```

6. Encoder/Decoder number of layers: پارامتر بعدی تعداد لایه‌های Transformer در دو بخش Enoder و Decoder می‌باشد که مشابه قبل برابر 6 می‌گذاریم. همچنین چون ساختار multihead attention می‌خواهیم داشته باشیم پس تعداد head هر لایه transformer را برابر با 8 در نظر می‌گیریم:

```
enc_layers: 6

dec_layers: 6

heads: 8
```

7. Drop Out: پارامتر مهم دیگر، تعیین احتمال dropout می‌باشد. خود این پارامتر در دو جای مختلف تعریف می‌شود: بعد از توابع فعال ساز که به آن dropout گفته می‌شود و در هنگام attention که به آن dropout attention می‌گوییم. این پارامتر از این جهت که مدل در حین آموزش، بیش از حد جملات مرجع را حفظ نکند و به اصطلاح overfit نشود، مهم است. مقدار هر دو آنها را برابر 0.2 در نظر می‌گیریم:

```
dropout_steps: [0]

dropout: [0.2]

attention_dropout: [0.2]
```

8. Learning Rate: همانطور که گفته شد تعیین learning rate بسیار حائز اهمیت است از آن جهت که مدل در step های متوالی بتواند یادگیری مناسبی داشته باشد. برای اینکه از فرمول مشابه مقاله برای تغییر learning rate استفاده شود در ابزار OpenNMT، متد decay را به صورت زیر تنظیم می‌کنیم. همچنین در اینجا warmup_steps را برابر با 800 در نظر می‌گیریم:

```
decay_method: "noam"

warmup_steps: 800
```

9. Weight Initialization: در آموزش شبکه‌های عصبی عمیق نحوه initialize کردن وزن های شبکه موثر است که در اینجا از روشی موسوم به Glorot initialization که استفاده از آن در transformerها مرسوم است، استفاده می‌کنیم:

```
param_init: 0
```

```
param_init_glorot: true
```

10. data type: بررسی‌های اخیر نشان می‌دهد که استفاده از half-precision floating point یا همان 16FP به جای 32FP در عین افزایش سرعت آموزش مدل و کاهش فضای اشغالی، تقریباً اثری روی عملکرد مدل ندارد بنابراین از نوع داده در این مدل استفاده می‌کنیم:

```
model_dtype: "fp16"
```

* درباره Scoring: در این ابزار هنگام آموزش، مقادیر accuracy و perplexity روی دیتاست آموزش و ارزیابی در هر step گزارش می‌شد.

Fixed parameters

در اینجا هم سه تا از اصلی‌ترین پارامترهای مدل که به دلیل محدودیت های موجود، امکان تغییر آنها وجود ندارد همان پارامترهای گفته شده برای ابزار FairSeq است بنابراین از توضیح مجدد جزیئات آن پرهیز می‌کنیم.

1. Encoder/Decoder number of layers
2. Encoder/Decoder Embedding size
3. Encoder/Decoder Feedforward Embedding Size

همچنین تعدادی پارامتر در ابزار OpenNMT هم وجود دارد که به دلیل محدودیت تعداد GPU قابل تنظیم نیست نظیر world_size که این پارامتر تعداد GPU های مورد استفاده در حین آموزش مدل می‌باشد که در صورتی که دسترسی به چندین GPU داشتیم می‌توانستیم به صورت موازی به آموزش مدل پرداخته و پارامتر batch_size و valid_batch_size را بزرگ‌تر در نظر بگیریم.

Sample Output for every 200 train steps

در ادامه همان سه جمله نمونه از دادگان تست که انتخاب کرده بودیم را به همراه ترجمه مدل بدست آمده از ابزار OpenNMT به ازای هر 200 تا train step را آورده‌ایم :

Example 1: the Israel defense forces said that more than 20 mortars and rockets were subsequently fired into their territory.

Reference : نیروهای دفاعی اسراییل گفتند بیش از 20 خمپاره و موشک متعاقبا به قلمرو انان شلیک شد.

400Hypothesis_check-point: او گفت: " شبه نظامیان جدید را در شهر بیرون آورد, و آن ها را ترک کردند.

600Hypothesis_check-point: وی گفت که ارتش اسرائیل همچنین بیش از 20 شبه نظامی را در دست داده اند.

800Hypothesis_check-point: اداره ارتش اسرائیل در 20 مه به منطقه ای گفتند و در نتیجه آن ها دست گرفتند.

1000Hypothesis_check-point: ارتش اسرائیل در انفجار 20 نفر کشته و زخمی ها را به تن داد.

1200Hypothesis_check-point: در این درگیری که اسرائیل کشته شده اند و گفت که نیروهای نظامی به خاک سپرده شده است.

1400Hypothesis_check-point: دان نیوز گزارش داد که انفجاری به ضرب گلوله کشت.

1800Hypothesis_check-point: اداره ارتش اسرائیل گفت که بیش از 20 تن را به جای آن ها حمله کردند.

Example 2: that "s how Washington dealt with the Soviet Union and China in the 1970 s and 80 s.

Reference: همین گونه بود که واشنگتن در دهه های 70 و 80 میلادی با جماهیر شوروی و چین کنار آمد.

400Hypothesis_check-point: این حزب در ماه ژانویه در منطقه جنوبی و پنج ماه گذشته در استان هلمند و پنج سال کشته شدند.

600Hypothesis_check-point: ایالات متحده به طور رسمی سوریه و چین برگزار شد.

800Hypothesis_check-point: به همین علت است که چین در دهه 1980 به پایان رسید.

1000Hypothesis_check-point: عجیب است که چرا کشورهای جنگ در سال های 1992 و 80 نفر باشد.

1200Hypothesis_check-point: این است که نخست وزیر اتحاد جماهیر شوروی و چین در دهه 1980 ترکیه شد.

1400Hypothesis_check-point: این بود که چگونه اتحاد جماهیر شوروی و شوروی در دهه 70 میلادی و 70 میلادی.

1800Hypothesis_check-point: به همین دلیل است که اتحادیه اروپا در سال 1970 و افغانستان هستند.

Example 3: officials killed two of the attackers, according to a regional police chief.

Reference :طبق گفته رئیس پلیس محلی, ماموران دو نفر از مهاجمین را کشته اند.

400Hypothesis_check-point: رسانه ها گزارش دادند که نیروهای امنیتی در یک ایست بازرسی در وزیرستان شمالی کشته شدند.

600Hypothesis_check-point: مقامات به گفته مسئولان, مقامات انتظامی منطقه باجور را متهم کرده اند.

800Hypothesis_check-point: بنا به گفته رسانه ها, مقامات پلیس در یک حمله زخمی شدند.

1000Hypothesis_check-point: بنا بر گزارش رسانه ها, دو مهاجم در یک حمله به پلیس در منطقه اوراکزی کشته شدند.

1200Hypothesis_check-point: مقامات گفتند که دو مهاجم بطور کلی در واکنش به رسانه ای در منطقه اوراکزی در حال حمله هستند.

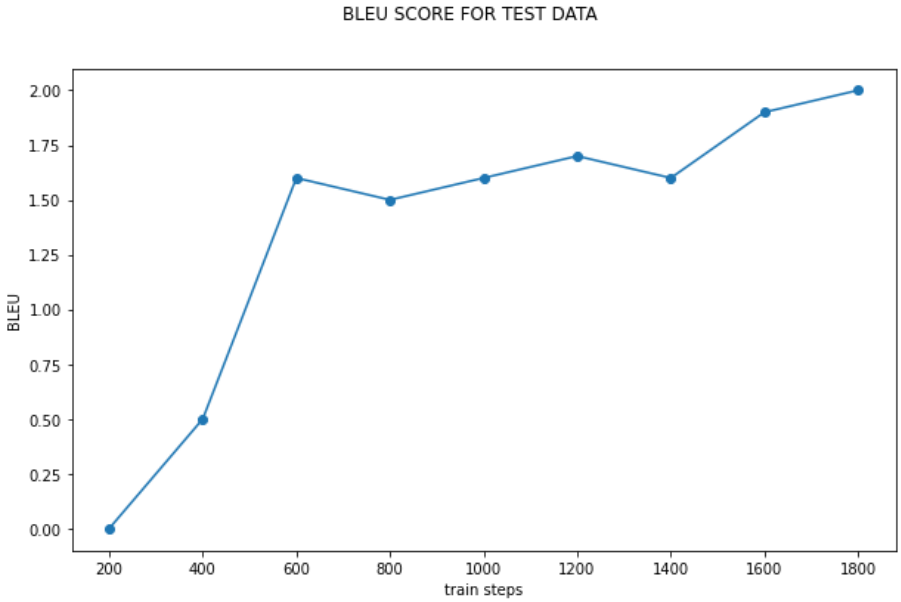
1400Hypothesis_check-point: مقامات محلی به رسانه ها گفتند که دو مهاجم در یک حمله به پلیس در منطقه اوراکزی کشته شدند.

1800Hypothesis_check-point: مقامات محلی به رسانه ها گفتند که دو نفر از پیکارجویان را در منطقه سوات کشته است.

Scoring Results[9]

در ادامه، ارزیابی از مدل طی 9 checkpoint مختلف که هر checkpoint شامل 200 step از آموزش می‌باشد، ارائه می‌دهیم :

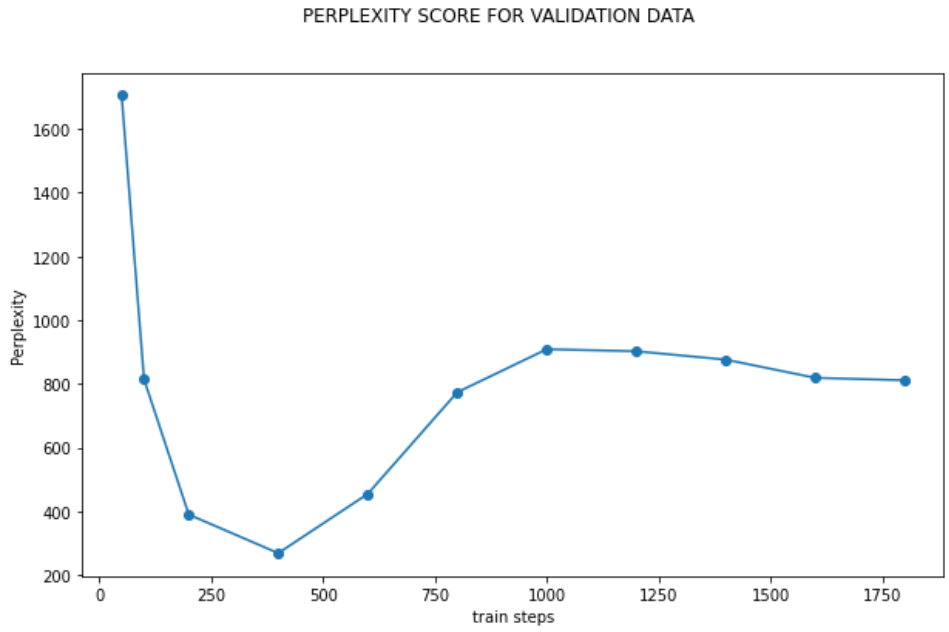
در اولین نمودار، معیار bleu را برحسب تعداد step برای دادگان تست نشان می‌دهد که تقریبا مشهود است که با افزایش step ها، معیار bleu رو به افزایش است که در نهایت به مقدار 2.02 می‌رسد. (اگر آموزش ادامه پیدا می‌کرد احتمالاً امتیاز bleu هم افزایش پیدا می‌کرد اما در مقابل perplexity هم احتمالاً افزایش می‌یافت که مطلوب ما نبود. بنابراین برای جلوگیری از overfit شدن مدل روی داده آموزش و... آموزش را در همین حد به پایان رساندیم یا اصطلاحاً early stopping به صورت دستی انجام دادیم.)



شکل 4 : نمودار امتیاز BLEU بر حسب train steps برای دادگان تست

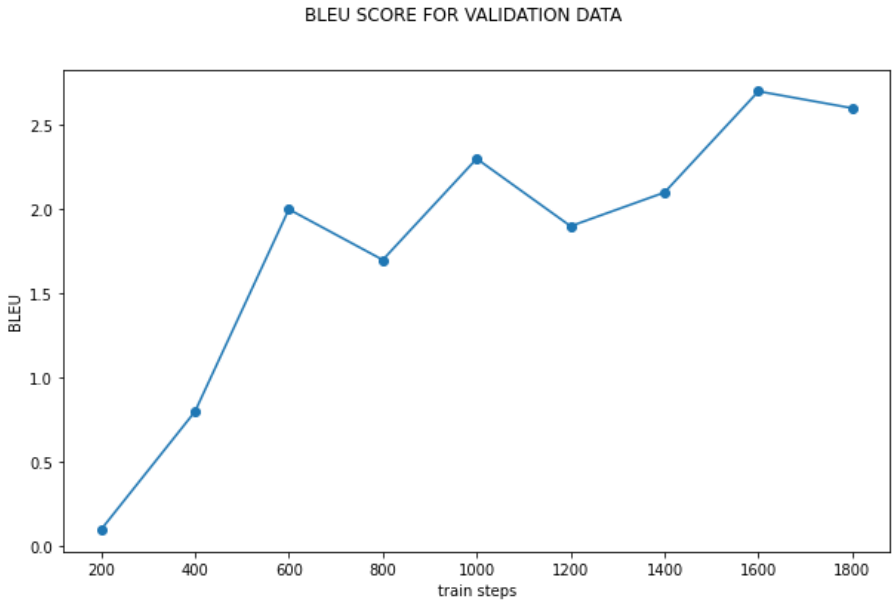
در نمودار دوم، معیار perplexity بر حسب تعداد step نمایش داده شده است. در ابتدا perplexity رو به کاهش است که نشان می‌دهد که جملات تولید شده از نظر زبانی درست هستند ولی با گذشت زمان و افزایش معیار bleu، مقدار perplexity افزایش پیدا می‌کند به این معنی که خروجی ترجمه شده با مرجع تعداد unigram های مشترک زیادی دارند ولی جمله بوجود آمده از نظر زبانی کمی ایراد داشته باشد. (به همین دلیل همانطور که گفته شد آموزش این مدل را بیشتر از 1800 step ادامه ندادیم.)

[9] خروجی مدل برای دیتاست تست در کنار این گزارش پیوست شده است.



شکل 5 : نمودار امتیاز Perplexity بر حسب train steps برای دادگان ارزیابی

در نهایت، نمودار bleu بر حسب تعداد step برای دادگان ارزیابی رسم شده است که مانند شکل 6، معیار bleu افزایش یافته است. (بیشترین مقدار آن برابر با 2.7 است.)



شکل 6: نمودار امتیاز BLEU بر حسب Epoch برای دادگان ارزیابی

NMT System Toolkits Evaluation metrics

اگر منظور مقایسه مدل‌های خروجی این دو ابزار است؛ برای ارزیابی مدل‌های ترجمه ماشینی معروف ترین معیار، معیار bilingual evaluation understudy یا به اختصار BLEU است. ایده اصلی این معیار این است که هر قدر ترجمه ماشین به ترجمه انسان نزدیک باشد بهتر است و امتیاز BLEU آن بیشتر خواهد بود. معیارهای مشابه دیگری نظیر Meteor نیز وجود دارند اما اغلب همین معیارBLEU معیار اصلی مقایسه مدل های NMT است.
معیار دیگری که می‌توان در نظر گرفت perplexity جملات خروجی سیستم است که به نوعی نشانگر این است که معیاری از این است که Language model سیستم چقدر می‌تواند جملات خوبی را تولید کند. (که البته لزوماً هم ممکن است ترجمه خوبی نباشد اما با این حال انتظار داریم جملات خروجی مدل از نظر زبانی جملات درستی باشند.)
معیار دیگری که در هنگام آموزش مدل می‌توان بررسی کرد مقدار loss یا accuracy است که با توجه به آن می‌توان مطمئن شد که هایپر پارامترهای مدل مناسب هستند و مدل در حال یادگیری است.

اگر منظور مقایسه خود ابزارها بدون در نظر گرفتن مدل خروجی است؛ می‌توان User friendly بودن و Ease of use را در نظر گرفت که برای config کردن مدل خروجی چه مقدار تنظیمات در دسترس است و نحوه تنظیم کردن این موارد چگونه است. مثلاً ابزار OpenNMT تعداد پارامترهای قابل تنظیم کمتری دارد نسبت به FairSeq اما در مقابل نحوه تنظیم کردن این پارامترها با استفاده از یک فایل configuration با فرمت yaml است که روش مناسب تری است نسبت به دادن پارامترها در ابزار FairSeq که به صورت آرگومان در command-line داده می‌شود. همچنین اینکه پیاده‌سازی مدل در این ابزارها به چه صورت است نیز در سرعت train مدل و نیز سرعت translation بسیار موثر است که در کاربردهای صنعتی این موضوع بسیار تعیین کننده است. مثلاً فریمورک Pytorch برای کاربردهای ریسرچ و فریمورک Tensorflow برای کاربردهای صنعتی مناسب تر است (scale پذیرتر است) و OpenNMT امکان پیاده‌سازی با هردو فریمورک ها را دارد اما برعکس FairSeq تنها از Pytorch استفاده می‌کند. همچنین marianNMT (که در صورت پروژه معرفی شده بود اما ما از آن استفاده نکردیم) چون پیاده‌سازی با استفاده از C++ است احتمالاً از نظر سرعت از هر دوی OpenNMT و FairSeq سریعتر است.
بنابراین با توجه به توضیحات فوق به نظر می‌رسد در مرحله ریسرچ کهflexibilty بیشتر در تنظیم هایپرپارامترها برای ما مهم است استفاده از FairSeq مناسب تر است. اما وقتی به مدل مطلوب رسیدیم؛ برای deploy کردن آن مدل برای کاربردهای صنعتی که سرعت و مقیاس پذیری در اولویت است استفاده از OpenNMT مناسبت تر است.