به نام خداوند جان و خرد

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

پردازش زبان طبیعی

تمرین شماره ۵

نام و نام خانوادگی: علی خرم فر

شماره دانشجویی: ۸۱۰۱۰۲۱۲۹

خردادماه ۱۴۰۳

فهرست مطالب

۱_ پاسخ سوال اول

ابتدا کتابخانه‌های مورد نظر نصب شده و آن‌ها را import می‌کنیم. حال یکی یکی بخش‌های مختلف سوال را بررسی می‌کنیم.

۱-۱_ دادگان

ابتدا دیتاست مربوط را از لینک ارائه‌شده دانلود کرده و پس از استخراج فایل فشرده‌شده zip ، لیستی از فایل‌های موجود از آن را خروجی می‌گیریم:

['LICENSE', 'README', 'MIZAN.en-fa.fa', 'MIZAN.en-fa.xml', 'MIZAN.en-fa.en']

تعداد کل خطوط و استخراج سه خط اول

به این منظور ۲ تابع پیاده‌سازی شد. یکی برای شمارش تعداد کل خطوط و دیگری برای برگرداندن محتوای سه خط اول. خروجی این توابع برای هر فایل به صورت زیر است :

انگلیسی:

تعداد کل خطوط : **۱۰۲۱۵۹۷**

1. The story which follows was first written out in Paris during the Peace Conference
2. from notes jotted daily on the march, strengthened by some reports sent to my chiefs in Cairo.
3. Afterwards, in the autumn of 1919, this first draft and some of the notes were lost.

فارسی:

تعداد کل خطوط : **۱۰۲۱۵۹۷**

۱. داستانی که از نظر شما می‌گذرد، ابتدا ضمن کنفرانس صلح پاریس از روی یادداشت‌هائی که به طور روزانه در حال خدمت در صف برداشته شده بودند
۲. و از روی گزارشاتی که برای رؤسای من در قاهره ارسال گردیده بودند نوشته شد.
۳. بعدا در پائیز سال ۱۹۱۹، این نوشته اولیه و بعضی از یادداشت‌ها، مفقود شدند.

باتوجه به اینکه هر جمله انگلیسی معادل فارسی دارد، پس تعداد خطوط برابرند.

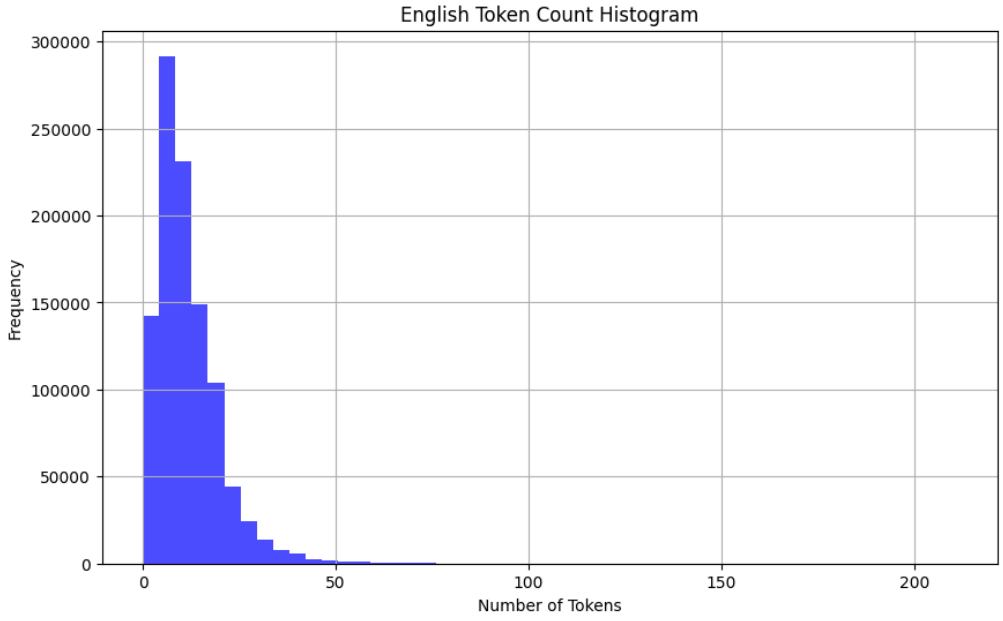
پس از این مرحله برای اینکه کنترل بهتری روی داده‌ها داشته باشیم آن‌ها را به قالبی دیتافریم تبدیل می‌کنیم.

```
data.head()
```

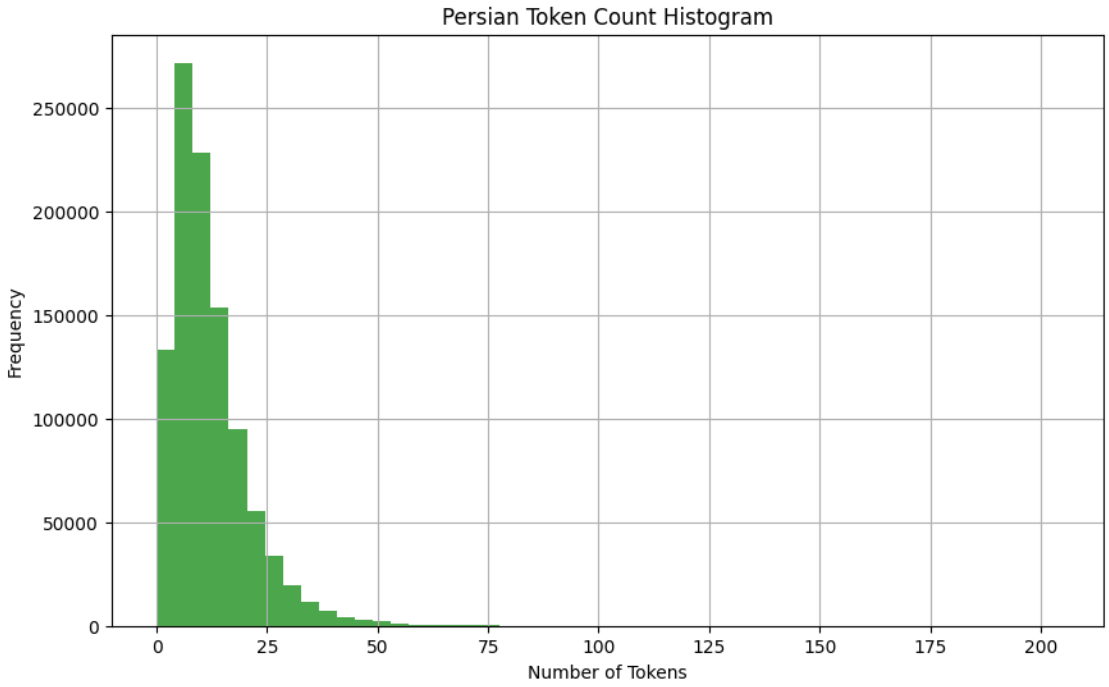
	English	Persian
0	The story which follows was first written out ...	داستانی که از نظر شما می‌گذرد، ابتدا ضمن کنفرا...
1	from notes jotted daily on the march, strength...	و از روی گزارشاتی که برای رؤسای من در قاهره ار...
2	Afterwards, in the autumn of 1919, this first ...	بعدا در پائیز سال 1919، این نوشته اولیه و بعضی...
3	It seemed to me historically needful to reprod...	به نظر من چنان می‌آمد که از نظر تاریخی رخدادها...
4	So it was built again with heavy repugnance in...	لذا این داستان مجددا با تفاوت زیادی، در زمستان...

هیستوگرام برای تعداد توکن‌ها

Whitespace به کاراکترهایی اشاره دارد که در متن به عنوان فاصله بین کلمات عمل می‌کنند. این کاراکترها شامل space، tab، و خط جدید هستند. در بیشتر زبان‌ها، کلمات با استفاده از این کاراکترها از هم جدا می‌شوند و به همین دلیل توکنایز کردن با استفاده از فاصله یکی از روش‌های متداول است. حال به بررسی هیستوگرام توکن‌های هر زبان خواهیم پرداخت:



شکل ۱ هیستوگرام توکن‌های انگلیسی



شکل ۲ هیستوگرام توکن‌های فارسی

در هیستوگرام های فارسی و انگلیسی مشاهده می‌کنیم که تعداد زیادی از سطرها دارای تعداد کمی توکن (کمتر از ۲۰) هستند. این دو هیستوگرام نشان می‌دهد که هر دو مجموعه داده‌های انگلیسی و فارسی دارای توزیع مشابهی در تعداد توکن‌ها هستند و اکثر جملات کوتاه هستند.

کاهش حجم دیتاست

برای فیلتر کردن مجموعه داده‌ها به منظور حذف سطرهایی که تعداد توکن آنها کمتر از ۱۰ و بیشتر از ۵۰ است یک دستور اجرا شد. جملات خیلی کوتاه و خیلی بلند که ممکن است باعث پیچیدگی و تنوع بیش از حد در داده‌ها شوند، حذف شدند. تعداد دادها قبل و بعد از فیلترکردن به صورت زیر است:

تعداد سطرها قبل از فیلتر کردن **۱۰۲۱۵۹۷** بوده و بعد از فیلتر کردن به **۵۴۸۱۸۵** کاهش یافت.

	English	Persian	English_Tokens	Persian_Tokens
0	The story which follows was first written out ...	داستانی که از نظر شما می‌گذرد، ابتدا ضمن کنفرا...	14	26
1	from notes jotted daily on the march, strength...	و از روی گزارشاتی که برای رؤسای من در قاهره ار...	17	15
2	Afterwards, in the autumn of 1919, this first ...	بعدا در پائیز سال 1919، این نوشته اولیه و بعضی...	16	14
4	So it was built again with heavy repugnance in...	لذا این داستان مجددا با تفاوت زیادی، در زمستان...	22	20
5	The record of events was not dulled in me and ...	من وقایع و رخدادهای اصلی را فراموش نکرده بودم،...	23	17

ابتدا با Random Seed برابر ۴۲ داده‌ها را شافل کردیم. برای اینکه نتیجه کار تایید شود مجددا از چند سطر اول دیتاست خروجی می‌گیریم:

	English	Persian
0	Yes, once or twice, when he came into Coombe T...	.بله، یک یا دو بار وقتی به کومب تریسی آمده بود
1	and yet he scanned with obstinate attention, t...	...با این همه، با دقتی لجوجانه، ظلماتی را که در آ
2	The next he knew, he was dimly aware that his ...	...وقتی به خود آمد، به نحو مبهمی متوجه شد که زبان
3	Dana drove as fast as she could to Theodore Ro...	...دنا با بیشترین سرعتی که می‌توانست به سوی مدرسه
4	abounds with incidents that fill the hearers w...	... مملو از حوادثی است که شنوندگان را مات و مجذوب

مشخص است که عمل شافل به خوبی انجام شده است.

سپس داده‌ها را به تعداد گفته شده از ابتدا Split می‌کنیم:

```
train_size = 500000
valid_size = 5000
test_size = 10000

train_data = shuffled_data.iloc[:train_size]
valid_data = shuffled_data.iloc[train_size:train_size + valid_size]
test_data = shuffled_data.iloc[train_size + valid_size:train_size + valid_size + test_size]
```

ذخیره فایل‌ها

باتوجه به اینکه برای این تمرین از محیط Kaggle استفاده شد، داده‌ها را در پوشه خروجی زیر ذخیره کردیم:

output_dir = '/kaggle/working/raw_data'/

پس از این مرحله با ذخیره فایل آن‌ها را بر روی محیط Kaggle ذخیره کردیم تا در اجراهای بعدی به راحتی قابل استفاده باشند. برای تایید نهایی لیست پوشه‌ها را اینبار از مسیر ورودی Kaggle خروجی می‌گیریم:

Saved files: ['valid.en', 'valid.fa', 'train.fa', 'test.fa', 'test.en', 'train.en']

```
!wc -l /kaggle/input/nlp-ca5/raw_data/train.fa
!wc -l /kaggle/input/nlp-ca5/raw_data/train.en
!wc -l /kaggle/input/nlp-ca5/raw_data/valid.fa
!wc -l /kaggle/input/nlp-ca5/raw_data/valid.en
!wc -l /kaggle/input/nlp-ca5/raw_data/test.fa
!wc -l /kaggle/input/nlp-ca5/raw_data/test.fa
```

```
500000 /kaggle/input/nlp-ca5/raw_data/train.fa
500000 /kaggle/input/nlp-ca5/raw_data/train.en
5000 /kaggle/input/nlp-ca5/raw_data/valid.fa
5000 /kaggle/input/nlp-ca5/raw_data/valid.en
10000 /kaggle/input/nlp-ca5/raw_data/test.fa
10000 /kaggle/input/nlp-ca5/raw_data/test.fa
```

۲-۱_ بخش اول : آموزش توکنایزر BPE و پیش پردازش دادگان

ابتدا فایل‌های ذخیره شده از مرحله قبلی را فراخوانی می‌کنیم.

توکنایزر BPE

در روش BPE ابتدا vocabs یا دایره ی واژگانی میسازیم که فقط شامل کاراکترهای به کار رفته در متون است. به عناصر این دا یره‌ی واژگان، توکن می‌گوییم. سپس دو توکنی که بیشتر از سایر توکن‌ها به صورت متوالی پشت سرهم آمده‌اند را به عنوان توکن جدید به دایره ی واژگان اضافه میکنیم. این اقدام را تا تعداد دلخواه تکرار می‌کنیم.

به این منظور ابتدا توکنایزر BPE را برای هر دو زبان انگلیسی و فارسی با استفاده از فایل‌های آموزشی مربوطه آموزش می‌دهیم. model_prefix نام پیشوند فایل مدل و vocab_size اندازه دایره واژگان را مشخص می‌کند. model_type نیز نوع مدل است که به BPE تعیین می‌کنیم.

```
# English
sentencepiece.SentencePieceTrainer.train(input=train_en_file, model_prefix='bpe_en', vocab_size=10000, model_type='bpe')
```

مدلی که آموزش داده‌ایم را در خروجی Kaggle ذخیره کرده تا بتوانیم در قسمت ۴ سوال از آن استفاده کنیم.

سپس فایل ورودی را توکنایز کرده و نتیجه را در فایل خروجی ذخیره می‌کنیم. به این منظور یک تابع پیاده‌سازی شد که تک تک خطوط را توکنایز کرده و خروجی را در **پوشه مربوطه** ذخیره می‌کند.

```
tokenize_data('/kaggle/input/nlp-ca5/raw_data/train.en', os.path.join(output_dir, 'train.en'), sp_en)
tokenize_data('/kaggle/input/nlp-ca5/raw_data/train.fa', os.path.join(output_dir, 'train.fa'), sp_fa)
tokenize_data('/kaggle/input/nlp-ca5/raw_data/valid.en', os.path.join(output_dir, 'valid.en'), sp_en)
tokenize_data('/kaggle/input/nlp-ca5/raw_data/valid.fa', os.path.join(output_dir, 'valid.fa'), sp_fa)
tokenize_data('/kaggle/input/nlp-ca5/raw_data/test.en', os.path.join(output_dir, 'test.en'), sp_en)
tokenize_data('/kaggle/input/nlp-ca5/raw_data/test.fa', os.path.join(output_dir, 'test.fa'), sp_fa)
```

output_dir = '/kaggle/working/tokenized_data'

فایل‌های توکنایزشده به صورت زیر هستند:

Tokenized files: ['train.fa', 'test.fa', 'en.vocab', 'valid.fa', 'test.en', 'fa.vocab', 'train.en', 'valid.en']

نمونه‌ای از توکن‌های تولیدشده:

```
!head /kaggle/input/nlp-ca5/tokenized_data/en.vocab -n 10
```

```
<unk>   0.0
<s>     0.0
</s>    0.0
_t      -0.0
he      -1.0
_a      -2.0
in      -3.0
_w      -4.0
_s      -5.0
_the    -6.0
```

توجه شود که باتوجه به اینکه فایل خروجی Kaggle را ذخیره کرده‌ایم در دستور بالا از فضای ورودی استفاده کرده‌ایم.

۵ خط اول توکنایزشده فارسی و انگلیسی نیز به صورت زیر هستند:

```
_Yes , _once _or _twice , _when _he _came _into _C oom be _Tr ace y .
_and _yet _he _scan ned _with _obst inate _attention , _the _darkness _in _which _he _walked
_The _next _he _knew , _he _was _dimly _aware _that _his _tongue _was _hur ting _and _that _he _was
_Dana _drove _as _fast _as _she _could _to _The od ore _R oose ve lt _Middle _School , _wondering _
_ab ounds _with _inc idents _that _fill _the _he are rs _with _wonder _and _astonishment ;
```

شکل ۳ نمونه داده توکنایزشده انگلیسی به روش BPE

```
_بله ، _یک _یا _دو _بار _وقتی _به _کو م ب _تریبس _آمده _بود
_با _د قتی _ل جو ج انه ، _ظل ماتی _را _که _در _آن _جای _داشت _مشاهده _می _کرد
_متوجه _شد _که _زبا نش _درد _دارد _و _او _را _با _وسیله _ای _به _سویی _می _برند
ولت _راند ، _در _دل _از _خودش _می _پرسید _چه _اتفاقی _ممکن _است _افتاده _باشد
، مملو _از _حوادثی _است _که _شنوندگان _را _مات _و _مجذوب _می _کند
```

شکل ۴ نمونه داده توکنایزشده فارسی به روش BPE

پیش‌پردازش با FairSeq

برای پیش‌پردازش داده‌های توکنایز شده از فرمان fairseq-preprocess استفاده می‌کنیم.

```
!fairseq-preprocess --source-lang en --target-lang fa \
    --trainpref /kaggle/input/nlp-ca5/tokenized_data/train \
    --validpref /kaggle/input/nlp-ca5/tokenized_data/valid \
    --testpref /kaggle/input/nlp-ca5/tokenized_data/test \
    --destdir /kaggle/working/fairseq_data \
    --workers 4 \
    --nwordssrc 10000 --nwordstgt 10000
```

source-lang en: زبان مبدا (انگلیسی)

target-lang fa: زبان مقصد (فارسی)

trainpref /kaggle/input/nlp-ca5/tokenized_data/train: مسیر فایل‌های آموزشی

validpref /kaggle/input/nlp-ca5/tokenized_data/valid: مسیر فایل‌های ارزیابی

testpref /kaggle/input/nlp-ca5/tokenized_data/test: مسیر فایل‌های تست

destdir /kaggle/working/fairseq_data: مسیر برای ذخیره داده‌های پیش‌پردازش شده.

nwordssrc 10000: اندازه دایره واژگان برای زبان مبدا ۱۰۰۰۰

nwordstgt 10000: دایره واژگان برای زبان مقصد ۱۰۰۰۰

این فرمان داده‌های توکنایز شده را به فرمت مناسب برای آموزش مدل‌های Fairseq تبدیل می‌کند. داده‌های متنی به فرمت باینری تبدیل می‌شوند تا سرعت و کارایی در مراحل آموزش مدل افزایش یابد.

فایل‌های تولیدشده نیز به صورت زیر هستند:

```
Preprocessed files: ['dict.en.txt', 'test.en-fa.fa.bin', 'train.en-fa.fa.idx', 'test.en-fa.en.bin', 'valid.en-fa.fa.bin', 'train.en-fa.fa.bin', 'dict.fa.txt', 'train.en-fa.en.bin', 'valid.en-fa.fa.idx', 'test.en-fa.en.idx', 'preprocess.log', 'valid.en-fa.en.idx', 'test.en-fa.fa.idx', 'valid.en-fa.en.bin', 'train.en-fa.en.idx']
```

dict.en.txt: دایره واژگان زبان انگلیسی.

dict.fa.txt: دایره واژگان زبان فارسی.

train.en-fa.en.bin و train.en-fa.en.idx: داده‌های آموزشی برای زبان انگلیسی

train.en-fa.fa.bin و train.en-fa.fa.idx: داده‌های آموزشی برای زبان فارسی

valid.en-fa.en.bin و valid.en-fa.en.idx: داده‌های ارزیابی برای زبان انگلیسی

valid.en-fa.fa.bin و valid.en-fa.fa.idx: داده‌های ارزیابی برای زبان فارسی

test.en-fa.en.bin و test.en-fa.en.idx: داده‌های تست برای زبان انگلیسی

test.en-fa.fa.bin و test.en-fa.fa.idx: داده‌های تست برای زبان فارسی

preprocess.log: فایل لاگ مربوط به فرآیند پیش‌پردازش

فایل‌های bin حاوی داده‌های واقعی و فایل‌های idx حاوی ایندکس‌های مرتبط با داده‌ها هستند.

۳-۱_ بخش دوم: آموزش مدل LSTM ENCODER-DECODER

برای حل این بخش از سوال پارامترهای مختلفی تست شد. ۲ مورد از تنظیمات که خروجی نسبتا خوبی داشتند را در اینجا بررسی می‌کنیم.

تحلیل نتایج آموزش با تنظیمات ۱

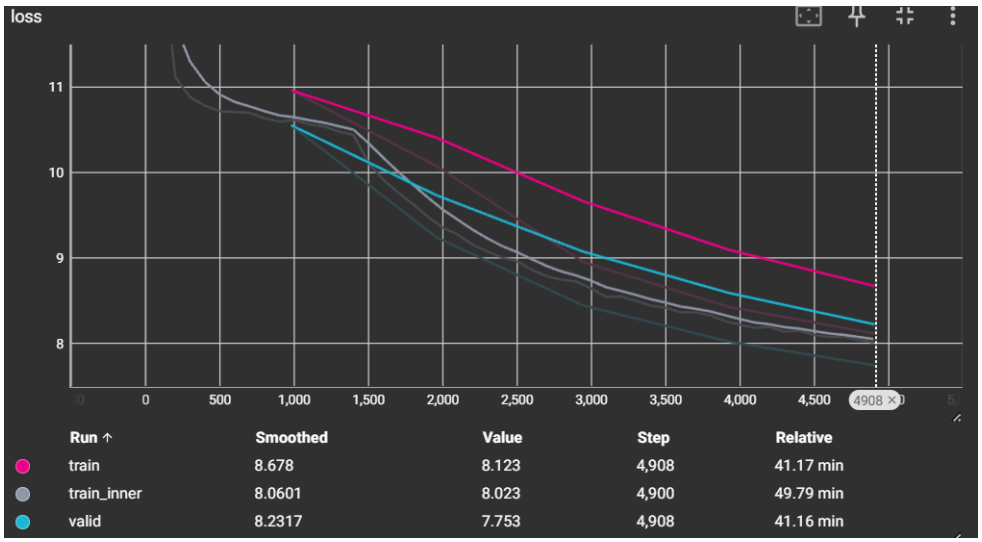
```
!fairseq-train \
    "/kaggle/input/nlp-ca5/fairseq_data" \
    --arch lstm --share-decoder-input-output-embed \
    --encoder-layers 6 --decoder-layers 6 \
    --optimizer adam --adam-betas '(0.9, 0.98)' --clip-norm 0.0 \
    --lr 2e-3 --lr-scheduler inverse_sqrt --warmup-updates 4000 \
    --dropout 0.3 --weight-decay 0.0001 \
    --criterion label_smoothed_cross_entropy --label-smoothing 0.2 \
    --max-tokens 8192 --batch-size 256 \
    --eval-bleu \
    --eval-bleu-args '{"beam": 5, "max_len_a": 1.2, "max_len_b": 10}' \
    --eval-bleu-detok moses \
    --eval-bleu-print-samples \
    --best-checkpoint-metric bleu --maximize-best-checkpoint-metric \
    --fp16 --memory-efficient-fp16 \
    --max-epoch 5 \
    --save-dir /kaggle/working/checkpoints/ \
    --tensorboard-logdir /kaggle/working/logs/
```

نسبتا به تنظیمات ارائه شده در فایل HandsOn تغییرات زیر اعمال شد:

batch-size 256: اندازه batch به تعداد ۲۵۶.

max-tokens 8192: حداکثر تعداد توکن‌ها در هر batch.

label-smoothing 0.2: تنظیم مقدار smoothing به ۰.۲.



شکل ۵ نمودار loss برای تنظیمات ۱ مدل LSTM

محور افقی نشان‌دهنده تعداد مراحل آموزش و محور عمودی نشان‌دهنده مقدار loss است.

منحنی صورتی (train): نشان‌دهنده loss داده‌های آموزشی است.

منحنی خاکستری (train_inner): نشان‌دهنده مقادیر loss داخلی در طول هر batch است.

منحنی آبی تیره (valid): نشان‌دهنده loss داده‌های ارزیابی است.

در ابتدای آموزش، مقدار loss برای هر دو داده‌های آموزشی و ارزیابی بالا بوده و در طول مراحل آموزش، هر دو منحنی کاهش می‌یابند. این نشان‌دهنده بهبود مدل در طول زمان و کاهش خطاها است با این حال، مقادیر loss همچنان بالا هستند.

خروجی BLEU روی داده‌های تست : ۴.۳۴

خروجی برای داده‌های Valid :

Run ↑	Smoothed	Value	Step	Relative
valid	4.0774	5.19	4,908	30.86 min

خروجی BLEU روی داده‌های ارزیابی : ۵.۱۹

تحلیل نتایج آموزش با تنظیمات ۲

```
!fairseq-train \
    "/kaggle/input/nlp-ca5/fairseq_data" \
    --arch lstm --share-decoder-input-output-embed \
    --encoder-layers 6 --decoder-layers 6 \
    --optimizer adam --adam-betas '(0.9, 0.98)' --clip-norm 0.1 \
    --lr 2e-3 --lr-scheduler inverse_sqrt --warmup-updates 4000 \
    --dropout 0.3 --weight-decay 0.01 \
    --criterion label_smoothed_cross_entropy --label-smoothing 0.2 \
    --max-tokens 8192 --batch-size 128 \
    --update-freq 4 \
    --eval-bleu \
    --eval-bleu-args '{"beam": 5, "max_len_a": 1.2, "max_len_b": 10}' \
    --eval-bleu-detok moses \
    --eval-bleu-print-samples \
    --best-checkpoint-metric bleu --maximize-best-checkpoint-metric \
    --fp16 --memory-efficient-fp16 \
    --max-epoch 5 \
    --num-workers 2 \
    --save-dir /kaggle/working/checkpoints_lstm/ \
    --tensorboard-logdir /kaggle/working/logs_lstm/
```

تغییرات پارامترها:

clip-norm 0.1--

batch-size 128--

update-freq 4--

مقدار ۴ یعنی بعد از جمع‌آوری ۴ batch، پارامترها به‌روزرسانی می‌شوند. این باعث می‌شود که مدل مانند استفاده از یک batch بزرگتر عمل کند

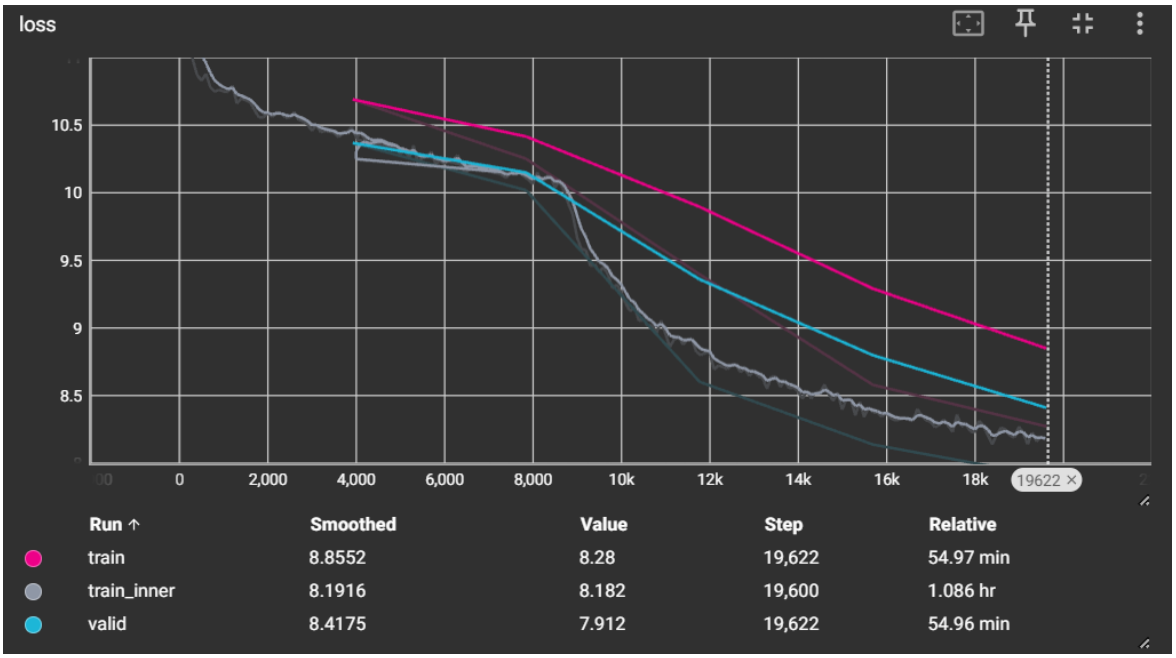
num-workers 2--

خروجی BLEU روی داده‌های تست : ۳.۸۷

خروجی برای داده‌های Valid :

Run ↑	Smoothed	Value	Step	Relative ↑
valid	3.7201	4.6	19,622	32.98 min

خروجی BLEU روی داده‌های ارزیابی : ۴.۶



شکل ۶ نمودار loss برای تنظیمات ۲ مدل LSTM

با توجه به اینکه مقدار BLEU در تنظیمات ۲ کمتر از تنظیمات ۱ است می‌توان نتیجه گرفت که عملکرد مدل در ترجمه بهبود نیافته است.

نحوه استفاده از پارامترها

پارامتر max-tokens

این پارامتر حداکثر تعداد توکن‌ها در هر batch را تعیین کرده و به Fairseq می‌گوید که تعداد کل توکن‌ها در یک batch نباید از مقدار تعیین شده بیشتر شود. برای مثال اگر مقدار max-tokens برابر با ۸۱۹۲ تنظیم شود، هر batch حاوی حداکثر ۸۱۹۲ توکن خواهد بود.

پارامتر batch-size

این پارامتر نیز تعداد نمونه‌ها یا جملات در هر batch را تعیین می‌کند. با محدود کردن تعداد توکن‌ها و تعداد جملات در هر batch، می‌توان تعادل مناسبی بین حجم داده‌ها و استفاده از منابع محاسباتی برقرار کرد. هرچند در چندجا نوشته شده بود که با افزایش بیش از حد اندازه توکن‌ها اندازه بچ نیازی به تغییر ندارد.

هدف از مشاهده نتایج ۲ تنظیم مختلف هم به همین دلیل بود و مقدارmax-tokens برابر با ۸۱۹۲ برای محدود کردن تعداد توکن‌ها و batch-size برابر با ۱۲۸ یا ۲۵۶ برای کنترل تعداد جملات استفاده شده است. هرچه که تعداد max-tokens را بیشتر کردیم دقت مدل افزایش پیدا می‌کرد(نسبت به ۴۰۰۰) ولی از طرفی فضای رم بیشتری از GPU نیاز داشت. باتوجه به منابع محدود برای انجام تمرین از این مقدار استفاده شد. از طرفی با افزایش بچ نیز سرعت افزایش می‌یافت.

به طور کلی محدود کردن تعداد توکن‌ها و جملات هر بچ به صورت همزمان کمک می‌کند که حافظه GPU بهینه‌تر استفاده شود و از تجاوز از ظرفیت حافظه جلوگیری شود و به صورت بهینه‌تری از منابع پردازشی استفاده کنیم.

۴-۱_ بخش سوم: آموزش مدل TRANSFORMER ENCODER-DECODER

تحلیل نتایج آموزش با تنظیمات ۱

نسبتا به تنظیمات ارائه شده در فایل HandsOn تغییرات زیر اعمال شد:

batch-size 256: اندازه batch به تعداد ۲۵۶.

max-tokens 8192: حداکثر تعداد توکن‌ها در هر batch.

label-smoothing 0.2: تنظیم مقدار smoothing به ۰.۲.

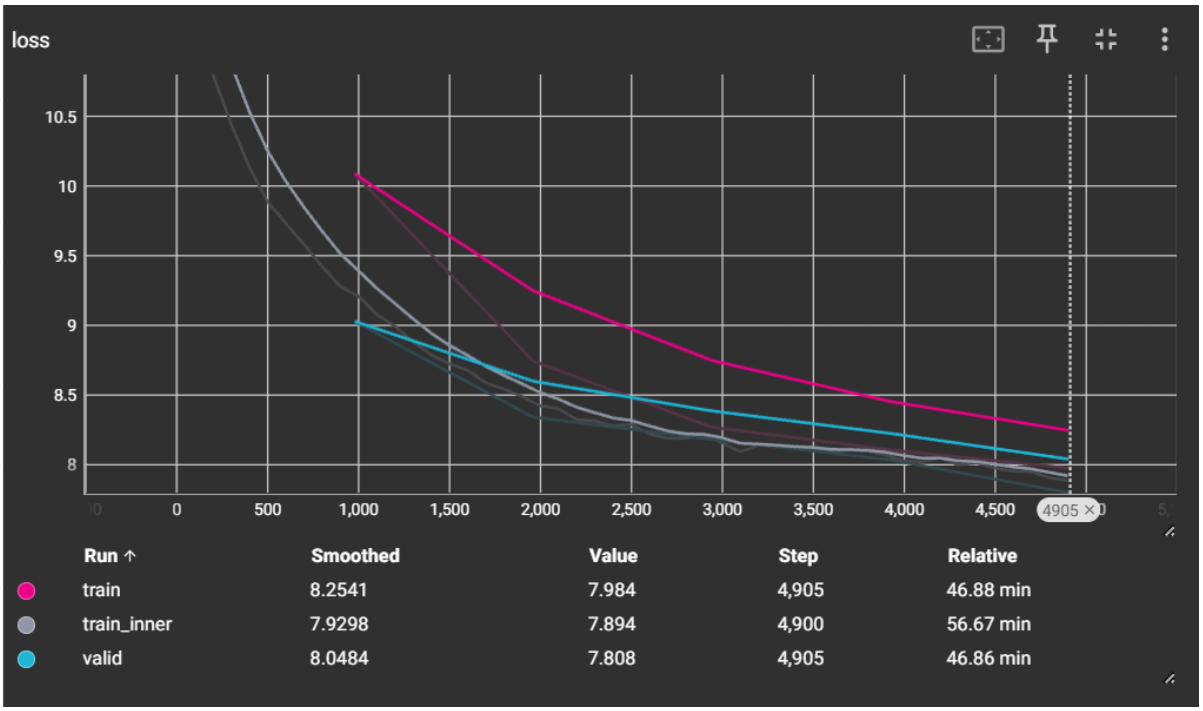
خروجی BLEU روی داده‌های تست : ۲.۷۸

مقدار BLEU به دست آمده ۲.۷۸ است که نسبتاً پایین است و نشان می‌دهد که مدل نیاز به بهبود دارد.

خروجی برای داده‌های Valid :

Run ↑	Smoothed	Value	Step	Relative
valid	2.5444	3.36	4,905	35.15 min

خروجی BLEU روی داده‌های ارزیابی : ۳.۳۶



شکل ۷ نمودار loss برای تنظیمات ۱ مدل Transformer

مقدار loss کاهش یافته با این حال، مقدار BLEU به دست آمده نشان می‌دهد که مدل نیاز به بهبود دارد. تنظیم بهتر پارامترها و یا افزایش ایپاک‌ها می‌تواند به بهبود عملکرد مدل کمک کند ولی به دلیل محدودیت منابع این امکان فراهم نشد.

تحلیل نتایج آموزش با تنظیمات ۲

تغییرات پارامترها:

clip-norm 0.1--

batch-size 128--

update-freq 4--

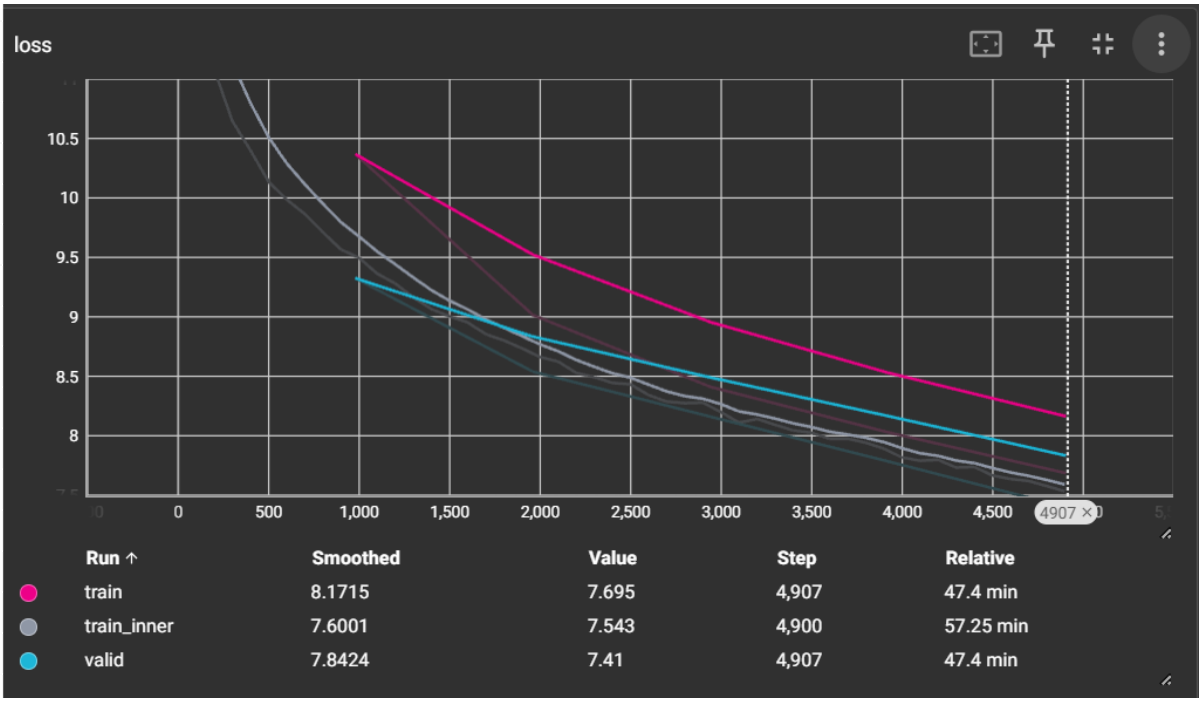
num-workers 2--

خروجی BLEU روی داده‌های تست : ۵.۵۵

خروجی برای داده‌های Valid :

Run ↑	Smoothed	Value	Step	Relative ↑
valid	4.314	6.31	4,907	35.55 min

خروجی BLEU روی داده‌های ارزیابی : ۶.۳۱



شکل ۸ نمودار loss برای تنظیمات ۲ مدل Transformer

همانطور که مشاهده می‌شود، نتیجه به مراتب بهتری با تنظیمات دوم به دست آمد که این مورد تاثیر پارامترهای تغییر یافته را نشان می‌دهد. کاهش نرخ یادگیری، افزایش weight decay کمک کرده است تا مدل عملکرد بهتری در داده‌های تست داشته باشد.

۵-۱_ بخش چهارم : معیار ارزیابی و بررسی داده ی تست

مقدار BLEU (Bilingual Evaluation Understudy) نمایانگر میزان نزدیکی ترجمه به مجموعه‌ای از ترجمه‌های انسانی با کیفیت خوب (که مجموعه مرجع خوانده می‌شود) است؛ بنابراین، با این روش نمی‌توان قابل فهم بودن ترجمه یا درستی آن از نظر دستوری را ارزیابی نمود. این روش برای ارزیابی ترجمه ماشینی در سطح کلی کاربرد دارد و در حالتی که برای ارزیابی تک تک جملات بکار برده شود، بسیار بد کار می‌کند. در این روش، n-gramهای تولید شده توسط مدل با ترجمه واقعی مقایسه می‌شوند و نسبت تشابه با یک عدد بیان می‌گردد.

برای هر دو مدل معیار BLEU بر روی داده‌های Valid در مرحله قبل گزارش شد.

فایل fairseq-generate خروجی گرفته شد و مقادی BLEU داده تست در مرحله قبل گزارش شد.

معیار ارزیابی COMET

COMET (Crosslingual Optimized Metric for Evaluation of Translation) یک فریمورک بر پایه شبکه عصبی برای ارزیابی ترجمه ماشینی (MT) است که به طور خاص برای پیش‌بینی کیفیت از دید زبان انسانی طراحی شده است. این معیار از مدل‌های چندزبانه از پیش‌آموزش دیده شده استفاده کرده تا مدل‌های ارزیابی ترجمه ماشینی را که با قضاوت‌های انسانی نزدیک هستند را آموزش دهد و به طور کلی ا با توجه به داده‌های انسانی فرایند آموزش آن‌ها انجام شده است. پس امتیازی که ارائه می‌شود با بلو متفاوت است. این معیار از میانگین هارمونیک بین فاصله تا مبدا و فاصله تا مرجع برای محاسبه کیفیت استفاده می‌کند.

COMET سه لیست از جملات را به عنوان ورودی دریافت می‌کند: جملات مبدا (source)، پیش‌بینی‌ها (hypothesis) و ترجمه‌های مرجع (reference). همچنین دو لیست از خروجی‌ها تولید می‌کند: لیست COMET Scores برای هر جمله ورودی و میانگین COMET Scores برای همه جملات ورودی که همان امتیاز کلی است. در مد ۲۲ استفاده شده که بهترین مدل فعلی است، Score ها بین ۰ تا ۱ قرار دارند.

استخراج و decode کردن جملات

مقایسه‌ی انجام شده در بخش ۴ تمرین بین دو مدل آموزش دیده شده توسط تنظیمات ۲ هستند.

فرمان fairseq-generate برای ارزیابی مدل ترجمه ماشینی استفاده می‌شود و فایل خروجی آن شامل جملات منبع، ترجمه مرجع و ترجمه ماشینی است. این فرمان را برای ۲ مدل قبلی اجرا کردیم و ۲ فایل مربوط به Tranformer_eval و LSTM_eval را بر روی حافظه Kaggle ذخیره کرده تا آن‌ها را در ادامه بررسی کنیم.

معیار بلو ارائه شده در قسمت‌های قبلی از خروجی همین فایل‌ها ارائه شد:

Generate test with beam=5: BLEU4 = 5.55, 30.3/8.6/3.1/1.2 (BP=1.000, ratio=1.026, syslen=191660, reflen=186853)

مدل‌های BPE آموزش‌داده شده برای انگلیسی و فارسی را که قبلا ذخیره کرده بودیم مجددا بارگذاری کرده و جملات را دیکد می‌کنیم.

برای استخراج نیز به صورت شرطی فایل را بررسی می‌کنیم:

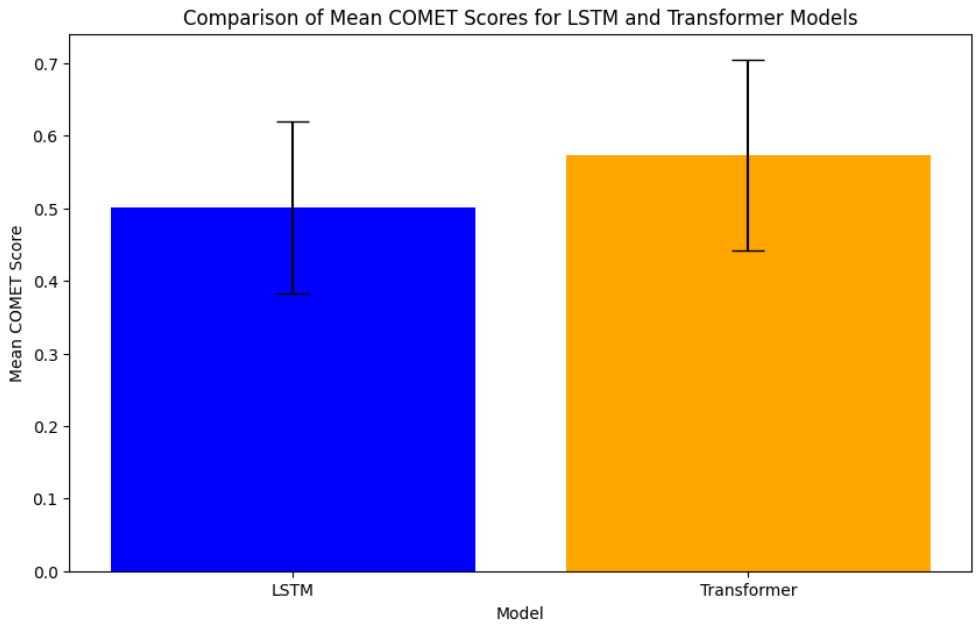
جملات منبع با S- شروع می‌شوند، ترجمه مرجع با T- و ترجمه ماشینی با H-.

محاسبه معیار COMET

رای ارزیابی نتایج داده‌های تست مدل‌های LSTM و Transformer که آموزش داده‌ایم، از کتابخانه unbabel-comet و مدل wmt22-comet-da استفاده می‌کنیم. پس از دانلود پیش‌بینی را بر روی داده‌های استخراج شده انجام می‌دهیم. پس از آن میانگین و انحراف معیار را محاسبه کرده و خروجی می‌گیریم:

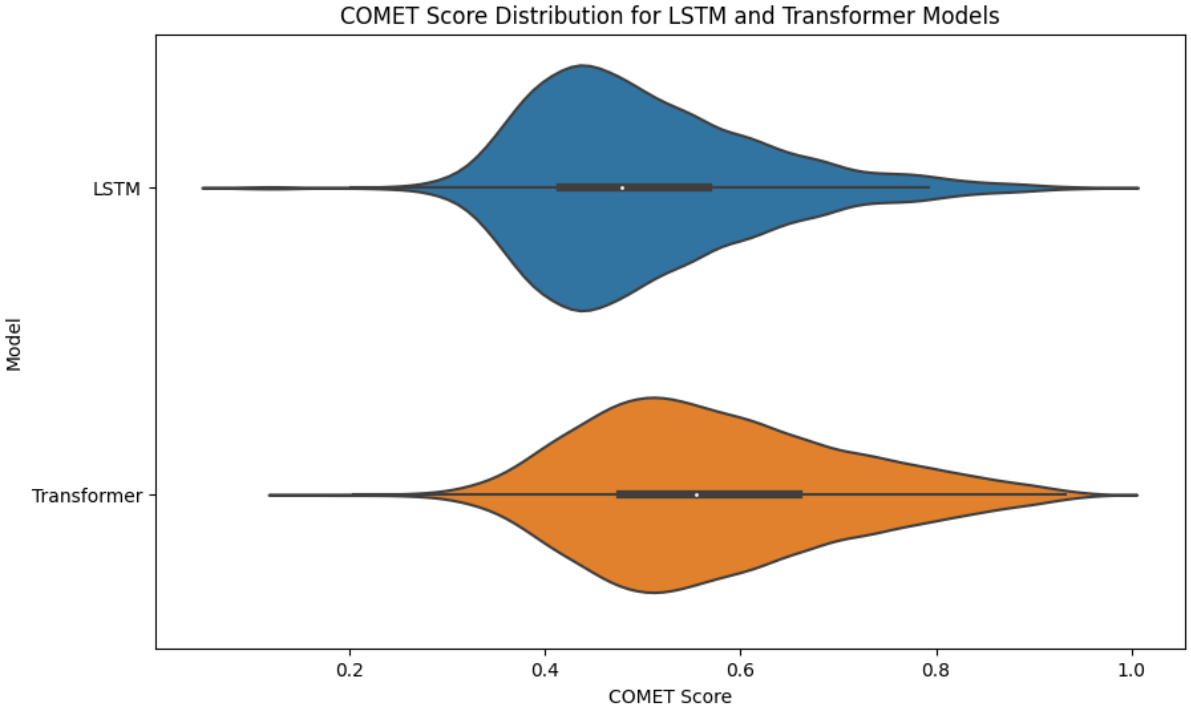
```
LSTM Model COMET Scores (Mean ± Std): 0.5013 ± 0.1190

Transformer Model COMET Scores (Mean ± Std): 0.5737 ± 0.1314
```



شکل ۹ نمودار مقایسه COMET برای مدل‌های LSTM و Transformer

نمودار بالا مقایسه‌ای بین میانگین امتیازات COMET برای دو مدل LSTM و Transformer را نشان می‌دهد. مدل Transformer عملکرد بهتری نسبت به مدل LSTM داشته است، زیرا امتیاز COMET بالاتری را به دست آورده است.



شکل ۱۰ نمودار توزیع COMET برای مدل‌های LSTM و Transformer

اثبات بهتربودن مدل Transformer با استنباط آماری

متاسفانه در نسخه آخر نوت بوک این مورد ثبت نشد. ولی در با کمک آزمون t-test و فرض صفر اینکه میانگین Transformer کمتر است، مقدار Pvalue نزدیک به صفر بدست آمده و اثبات شد که میانگین Transformer بیشتر است.

مقایسه نتایج COMET و BLEU

نتایج BLEU

مدل LSTM:

BLEU: 3.87

مدل Transformer:

BLEU: 5.55

نتایج COMET

مدل LSTM:

۰.۵۰۱۳

مدل Transformer:

۰.۵۷۳۷

امتیاز بالاتر BLEU و COMET برای مدل Transformer نسبت به LSTM نشان‌دهنده این است که مدل Transformer نه تنها در حفظ دقت و ساختار جملات بهتر عمل کرده، بلکه کیفیت کلی ترجمه‌ها نیز بهبود یافته است و امتیاز پایین BLEU نشان‌دهنده عملکرد ضعیف در ترجمه جملات به صورت دقیق است. به طور کلی استفاده از هر دو معیار BLEU و COMET کمک می‌کند تا تحلیل جامع‌تری از عملکرد مدل‌ها به دست آوریم. BLEU بیشتر به دقت و ساختار جملات توجه دارد، در حالی که COMET کیفیت کلی ترجمه‌ها را با توجه به مدل زبان انسانی ارزیابی می‌کند.

نکته دیگر این است که با توجه به اینکه امتیاز COMET بالای ۰.۵ نشان‌دهنده کیفیت قابل قبول ترجمه‌ها است، اما امتیاز BLEU نسبتاً پایین است، می‌توان نتیجه گرفت که مدل LSTM ممکن است در حفظ دقت و ساختار جملات مشکل داشته باشد، حتی اگر کیفیت کلی ترجمه‌ها مناسب باشد.