به نام خدا

دانشگاه تهران
دانشکدگان فنی
دانشکده مهندسی برق و کامپیوتر

درس پردازش زبان طبیعی

پاسخ تمرین ۵

نام و نام خانودگی: سامان اسلامی نظری

شماره دانشجویی: ۸۱۰۱۹۹۳۷۵

خرداد ماه ۱۴۰۳

فهرست

سوال اول

بخش اول: آموزش توکنایزر BPE و پیش‌پردازش دادگان

پس از آنکه توکنایزر BPE آموزش داده شد، دو فایل vocab. و model. برای هر دو زبان فارسی و انگلیسی تولید شدند. ابزار fairseq-preprocess با استفاده از این توکنایزر آموزش‌داده‌شده، داده‌های موجود را پیش پردازش می‌کند. خلاصه‌ای از مواردی که انجام می‌دهد شامل زیر است:

- توکنایز کردن: ابتدا داده‌ها را بر اساس توکنایزر داده‌شده توکنایز می‌کند.
- Subword segmentation: معمولا به منظور هندل کردن داده‌های خارج از مجموعه لغات و کاهش اندازه مجموعه لغات از این کار استفاده می‌شود.
- تبدیل توکن‌ها به اعداد: توکن‌ها را بر اساس موقعیتشان در مجموعه لغات به اعداد نگاشت می‌کند.

در کل پس از استفاده از fairseq-preprocess فایل‌های زیر تولید می‌شوند:

- فایل‌های دیتای باینری‌شده: این فایل‌ها شامل داده‌های توکنایز شده و ایندکس‌شده می‌باشد که به فرمت باینری ذخیره شده‌اند؛ به طوری که Fairseq به صورت بهینه می‌تواند از آن‌ها در مرحله آموزش استفاده کند.
  - فایل‌های زبان مبدا که انگلیسی است به صورت train.en.bin و test.en.bin و valid.en.bin می‌باشند.
  - به صورت مشابه فایل‌های زبان مقصد به صورت train.fa.bin و test.fa.bin و valid.fa.bin می‌باشند.
- فایل‌های ایندکس: در کنار فایل‌های باینری مذکور، یک سری فایل ایندکس با اکستنشن idx. نیز وجود دارند که برای دسترسی سریع‌تر و بهینه‌تر به داده‌ها مورد استفاده قرار می‌گیرند.
- فایل‌های ووکب: مجموعه لغات (ووکب) هر زبان در فایل dict.[lang].txt ذخیره شده است.

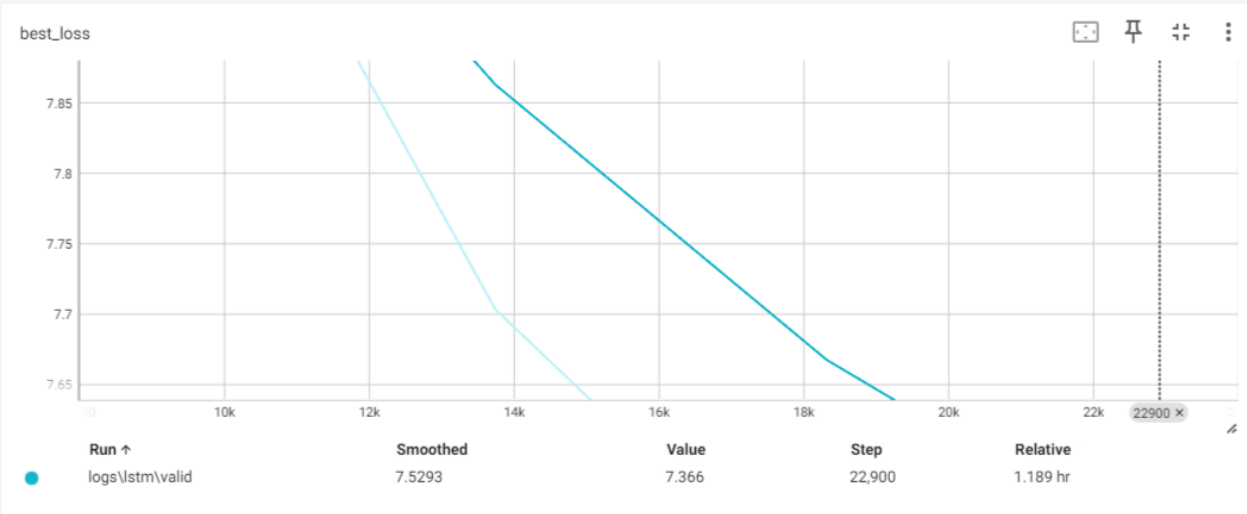
بخش دوم: آموزش مدل LSTM Encoder-Decoder

با استفاده از دستور زیر مشخصات معماری خواسته‌شده را آموزش می‌دهیم:

```
fairseq-train data-bin \
  --arch lstm --encoder-bidirectional \
  --encoder-layers 6 --decoder-layers 6 \
  --optimizer adam --adam-betas '(0.9, 0.98)' --lr 0.001 \
  --max-tokens 4000 \
  --criterion label_smoothed_cross_entropy --label-smoothing 0.2 \
  --save-dir checkpoints/lstm \
  --tensorboard-logdir logs/lstm \
  --max-epoch 5
```

هر دو آپشن batch-size و max-tokens اندازه هر batch را مشخص می‌کنند؛ مطابق با این مباحثه گاها batch-size به تعداد جملات موجود در یک batch اشاره می‌کند و هر جمله ممکن است از تعداد متفاوتی توکن تشکیل شده باشد. اما max-tokens دقیقا تعیین می‌کند که هر batch چند توکن باید داشته باشد؛ در این صورت هر تعداد جمله که نیاز باشد در یک batch جا می‌دهد تا به max-tokens برسیم.

این آپشن به ما کمک می‌کند که به نسبت سخت‌افزار در دسترس مرحله آموزش را پیش ببریم. مقدار ۴۰۰۰ به عنوان مقدار اولیه max-tokens انتخاب شد که با توجه به سخت‌افزار در دسترس مقدار مناسبی بود. همچنین دقت کنید که درست است که مقدار کم‌تر max-tokens باعث استفاده کم‌تر از سخت‌افزار می‌شود، اما باعث طولانی‌تر شدن زمان آموزش نیز خواهد شد. بنابراین انتخاب یک مقدار مناسب برای این آپشن بسیار حیاتی و مهم می‌باشد.



عکس ۱ مقادیر loss برای مدل lstm

بخش سوم: آموزش مدل Transformer Encoder-Decoder

با استفاده از دستور زیر، مدکل مذکور را آموزش دادیم:

```
fairseq-train data-bin \
  --arch transformer --encoder-layers 6 --decoder-layers 6 \
  --optimizer adam --adam-betas '(0.9, 0.98)' --lr 0.001 \
  --max-tokens 4000 \
  --criterion label_smoothed_cross_entropy --label-smoothing 0.2 \
  --save-dir checkpoints/transformer \
  --tensorboard-logdir logs/transformer \
  --max-epoch 5
```

بخش چهار: معیار ارزیابی و بررسی داده تست

نتایج مقدار BLEU به صورت زیر می‌باشد:

```
Generate test with beam=5: BLEU4 = 6.05, 33.2/10.0/4.0/1.7 (BP=0.877, ratio=0.884, syslen=199145, reflen=225396)
```

عکس ۲ مقدار BLEU برای مدل LSTM

ارزیابی Comet نیز برای هر مدل ابتدا باید hypothesis که همان ترجمه ماشینی است، source که جمله مبدا است و target که جمله رفرنس برای مقایسه با hypothesis است را جدا کرده و در فایل‌های مجزا قرار دهیم. سپس این سه مورد را برای هر ترجمه به مدل Comet می‌دهیم؛ برای هر ترجمه یک امتیاز داده و در نهایت یک امتیاز کلی برای تمام ترجمه‌ها (در این جا ۱۰ هزار ترجمه تست داشتیم) به ما ارائه می‌دهد:

- امتیاز کلی lstm: 0.7464
- امتیاز کلی transformer: -

برای توضیح عملکرد COMET به وبسایت unbabel مراجعه می‌کنیم:

COMET یا Crosslingual Optimized Metric for Evaluation of Translation نوعی بستر برای آموزش مدل‌های مبتنی بر شبکه‌های عصبی برای ارزیابی مدل‌های ترجمه ماشینی است. COMET برای این طراحی شده تا امتیاز انسان‌ها روی ترجمه‌های ماشینی را پیشبینی کند. این پیشبینی می‌تواند به ما در اتوماتیک کردن روند ارزیابی ترجمه ماشینی کمک کند.

چند تا از موارد کلیدی در COMET به شرح زیر می‌باشند:

- بر اساس شبکه‌های عصبی کار می‌کند؛ معمولا با استفاده از فاین تیون کردن شبکه‌های دیگر سعی می‌شود تا ارزیابی‌های انسانی را تقلید و پیشبینی کند.
- بر خلاف معیارهای دیگر که صرفا ترجمه انجام‌شده را با یک ترجمه رفرنس مقایسه می‌کنند، این روش جمله مبدا را نیز در نظر گرفته که باعث می‌شود کانتکست قوی‌تری برای ارزیابی داشته باشد.