گزارش ترجمه ماشینی

موضوع پروژه‌ی ترجمه ماشینی من تبدیل شعر به نثر بود. دیتایی که استفاده کردم از دیوان حافظ بود که تو گیت‌هاب گذاشتم(دیوان.pdf)

برای دیتای نثر همین پی‌دی‌اف بالا را با استفاده از google docs اوسیار کردیم ولی از آن‌جایی که پر از غلط بود مجبور شدیم تمام دیتا را یکبار چک کنیم. کد Data.py صفحاتی که مربوط به نثر هستند را جدا می‌کند و حاشیه‌ی آن را حذف می‌کند و هر ۱۰ عدد عکس را در یک پی‌دی‌اف می‌ریزد که بتوان در google docs اوسیار کرد؛ سپس کد 1.py دیتایی که از گوگل داکس دانلود کردیم (tarjome_hafez.txt) و ترجمه‌ی مربوط به هر بیت را با استفاده از شماره‌ای که قبل از آن نوشته شده جدا می‌کند و حاصل را در فایل nasr.txt می‌ریزد.

برای دیتای شعر سایت ganjoor.net را با استفاده از کد scrapy.py که زدیم کراول کردیم و اشعار را از آن‌جا برداشتیم. فایلی که این اسکریپت تولید می‌کند sher.txt است.

در آخر کد 2.py هر دو فایل sher.txt و nasr.txt را به فرمت مناسب برای opennmt در می‌آورد که حاصل آن f1.txt و f2.txt است.

برای validation از داده‌های ورودی ۱۹۰ عدد را انتخاب کردیم و بقیه ۴۰۰۰ تا را برای train استفاده کردیم. (۱۹۰ عدد از f1.txt و f2.txt را در f1_v.txt و f2_v.txt ریختیم)

برای کار با opennmt از دستورات زیر استفاده کردیم:

```
python preprocess.py -train_src data/f1.txt -train_tgt data/f2.txt -valid_src data/f1_v.txt -valid_tgt data/
f2_v.txt -save_data data_demo
```

سپس کامند زیر را می‌زنیم:

```
python train.py -data data/demo -save_model demo-model
```

بعد از زدن کامندهای بالا مدل شروع به یادگیری می‌کند ما در این‌جا اسم مدل‌مان را demo.model گذاشتیم.

عکسی از مدل در حال یادگیری:

```
[2018-07-11 11:21:01,109 INFO] Start training...
[2018-07-11 11:21:01,273 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 11:23:55,667 INFO] Step 50, 100000; acc:   4.96; ppl: 3100.58; xent:   8.04; lr: 1.00000; 306 / 503 tok/s;    174 sec
[2018-07-11 11:24:40,302 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 11:26:49,808 INFO] Step 100, 100000; acc:   6.27; ppl: 1318.68; xent:   7.18; lr: 1.00000; 210 / 410 tok/s;    349 sec
[2018-07-11 11:28:36,721 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 11:29:54,571 INFO] Step 150, 100000; acc:   2.39; ppl: 1761.27; xent:   7.47; lr: 1.00000; 278 / 373 tok/s;    533 sec
[2018-07-11 11:31:57,925 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 11:32:32,831 INFO] Step 200, 100000; acc:   5.62; ppl: 854.18; xent:   6.75; lr: 1.00000; 311 / 538 tok/s;    692 sec
[2018-07-11 11:35:10,218 INFO] Step 250, 100000; acc:   7.21; ppl: 830.02; xent:   6.72; lr: 1.00000; 282 / 492 tok/s;    849 sec
[2018-07-11 11:35:16,679 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 11:37:54,468 INFO] Step 300, 100000; acc:   5.65; ppl: 793.38; xent:   6.68; lr: 1.00000; 293 / 524 tok/s;   1013 sec
[2018-07-11 11:38:41,173 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 11:40:32,204 INFO] Step 350, 100000; acc:   7.61; ppl: 618.20; xent:   6.43; lr: 1.00000; 336 / 507 tok/s;   1171 sec
[2018-07-11 11:42:08,731 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 11:43:18,414 INFO] Step 400, 100000; acc:   7.71; ppl: 584.27; xent:   6.37; lr: 1.00000; 229 / 356 tok/s;   1337 sec
[2018-07-11 11:45:33,705 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 11:46:05,287 INFO] Step 450, 100000; acc:   9.19; ppl: 524.50; xent:   6.26; lr: 1.00000; 186 / 340 tok/s;   1504 sec
[2018-07-11 11:48:38,104 INFO] Step 500, 100000; acc:   9.53; ppl: 514.15; xent:   6.24; lr: 1.00000; 366 / 488 tok/s;   1657 sec
[2018-07-11 11:48:54,374 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 11:51:56,541 INFO] Step 550, 100000; acc:  14.06; ppl: 355.73; xent:   5.87; lr: 1.00000; 244 / 381 tok/s;   1855 sec
[2018-07-11 11:53:05,486 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 11:55:20,449 INFO] Step 600, 100000; acc:  14.86; ppl: 348.21; xent:   5.85; lr: 1.00000; 296 / 347 tok/s;   2059 sec
[2018-07-11 11:56:54,600 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 11:57:58,826 INFO] Step 650, 100000; acc:  12.99; ppl: 331.57; xent:   5.80; lr: 1.00000; 266 / 345 tok/s;   2218 sec
[2018-07-11 12:00:52,932 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 12:01:23,251 INFO] Step 700, 100000; acc:  14.80; ppl: 266.62; xent:   5.59; lr: 1.00000; 242 / 417 tok/s;   2422 sec
[2018-07-11 12:04:51,872 INFO] Step 750, 100000; acc:  16.06; ppl: 242.73; xent:   5.49; lr: 1.00000; 262 / 507 tok/s;   2631 sec
[2018-07-11 12:05:15,396 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 12:08:00,600 INFO] Step 800, 100000; acc:  14.78; ppl: 276.74; xent:   5.62; lr: 1.00000; 305 / 372 tok/s;   2819 sec
[2018-07-11 12:09:10,044 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 12:11:09,439 INFO] Step 850, 100000; acc:  18.05; ppl: 207.30; xent:   5.33; lr: 1.00000; 209 / 396 tok/s;   3008 sec
[2018-07-11 13:24:36,510 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 13:25:30,970 INFO] Step 900, 100000; acc:  17.33; ppl: 206.64; xent:   5.33; lr: 1.00000; 327 / 501 tok/s;   7470 sec
[2018-07-11 13:27:59,301 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 13:28:14,511 INFO] Step 950, 100000; acc:  16.65; ppl: 177.11; xent:   5.18; lr: 1.00000; 321 / 508 tok/s;   7633 sec
[2018-07-11 13:31:18,623 INFO] Step 1000, 100000; acc:  17.47; ppl: 170.68; xent:   5.14; lr: 1.00000; 154 / 311 tok/s;   7817 sec
[2018-07-11 13:31:50,875 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 13:34:36,456 INFO] Step 1050, 100000; acc:  18.80; ppl: 140.22; xent:   4.94; lr: 1.00000; 224 / 405 tok/s;   8015 sec
[2018-07-11 13:35:59,427 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 13:37:57,161 INFO] Step 1100, 100000; acc:  20.22; ppl: 123.43; xent:   4.82; lr: 1.00000; 291 / 422 tok/s;   8216 sec
[2018-07-11 13:39:44,917 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 13:40:37,967 INFO] Step 1150, 100000; acc:  19.57; ppl: 125.65; xent:   4.83; lr: 1.00000; 328 / 459 tok/s;   8377 sec
[2018-07-11 13:43:43,242 INFO] Loading train dataset from data_demo.train.pt, number of examples: 3998
[2018-07-11 13:43:56,736 INFO] Step 1200, 100000; acc:  20.43; ppl: 105.65; xent:   4.66; lr: 1.00000; 230 / 334 tok/s;   8575 sec
[2018-07-11 13:47:10,498 INFO] Step 1250, 100000; acc:  19.58; ppl:  99.37; xent:   4.60; lr: 1.00000; 293 / 455 tok/s;   8769 sec
```

ابزار opennmt نیاز به ۱۰۰۰۰۰ step دارد که یادگیری آن بیش‌تر از ۳ روز طول می‌کشد ولی بعد از هر ۵۰۰۰ step یک مدل به عنوان checkpoint ذخیره می‌کند. برای همین ما بعد از ۲ checkpoint یعنی یک دهم از مرحله‌ی یادگیری، خروجی آن را استفاده کردیم. و به همین دلیل مدل ما دقت خیلی بالایی ندارد.

Accuracy ~ 0.8 (برای داده‌ی train)

Perplexity ~ 2

Accuracy ~ 0.2(برای داده‌ی test)

Perplexity ~ 1400

برای گرفتن یک مثال از خروجی از کامند زیر استفاده کردیم:

```
python translate.py -model demo-model_step_10000.pt -src data/test.txt -output pred.txt -replace_unk -verbose
```

البته قبل از آن بیت زیر از حافظ را در فایل test.txt ریختیم .
(الا یا ایها الساقی ادر کأسا و ناولها که عشق آسان نمود اول ولی افتاد مشکل‌ها)
خروجی:

```
PRED 1: هان ای ساقی جام شراب را به چرخش در بده و به سرای آن می گویم که همصحبتی با عشق او در را می شکند
PRED SCORE: -11.7675
PRED AVG SCORE: -0.4707, PRED PPL: 1.6011
```