I. Оценка качества диффузии

Безусловная генерация

С твоим чекпоинтом получились следующие метрики:

1) 100 примеров

```
Run summary:
   rocstories-test/div 0.61755
   rocstories-test/mauve 0.86757
   rocstories-test/ppl 26.52681
```

2) 5000 примеров (дефолтное значение)

```
Run summary:
   rocstories-test/div 0.1352
   rocstories-test/mauve 0.71312
   rocstories-test/ppl 30.60526
```

3) 10000 примеров

```
Run summary:
   rocstories-test/div 0.08848
   rocstories-test/mauve 0.69827
   rocstories-test/ppl 30.33021
```

Условная генерация

Здесь я взял все тот же чекпоинт (то есть тот, который не предназначен для условной генерации) и протестировал его для условной генерации (еще без условного берта, просто чтобы было от чего отталкиваться)

```
Run summary:
   rocstories-test/bert-score 0.52846
   rocstories-test/bleu 0.00252
   rocstories-test/rouge1 0.117
   rocstories-test/rouge2 0.00611
   rocstories-test/rougeL 0.08938
```

II. Обучение энкодера

При обучении сразу не получилось нормально обучить модель, в итоге попробовал несколько стратегий обучения:

1) Обучать стандартно – сразу сэмплируем из $t \in [0,1]$. Получилось, что модель практически не учится. При стартовом лоссе в ~0.7, он опускался до ~0.69 в начале первой эпохи (то есть мог опуститься до 0.68 или подняться до 0.71 – но в среднем 0.69) и дальше практически не изменялся. Пробовал разные learning rate – все равно такое же поведение. Возможно, причина этого в том, что когда эмбеддинги очень близки к чистому шуму – то модели сложно отличить корректность текста, и, по сути, происходит обучение на шуме.

2) Исходя из логики, что за предсказания на шуме и на почти чистых эмбеддингах – штрафуем модель одинокого, попробовал добавить веса к лоссу: 1) Просто линейные (1 –

t) – то есть чем ближе к шуму, тем меньше штрафуем 2) Линейные, но с нормализацией. Результат остался примерно такой же, лосс застревал в районе 0.69.

3) Curricular t learning. Идея такая, что не сразу начинаем сэмплировать $t \in [0,1]$, а сначала обучим модель модель на $t \in [0,X]$, где $X < 1$, и далее увеличиваем X, пока не дойдем до 1. То есть, модель будет постепенно учится на более зашумленных картинках. Это сработало, итого, я обучил модель на 10 warmup эпохах (то есть за 10 эпох подняли шум от eps до 1) и еще на 5 полноценных эпохах. Получилось такое качество (accuracy считается как количество верных предсказаний модели по входному тексту и зашумленным эмбеддингам):

```
Run summary:
     train accuracy 0.75758
     train loss 0.41731
     valid accuracy 0.7374
     valid loss 0.45906
```

III. Оценка диффузии с условным бертом для генерации

Протестировал несколько classifier guidance scale, получились такие результаты:

Guidance Scale	BERT-Score	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
Без классификатора	0.52846	0.00252	0.11700	0.00611	0.08938
0.5	0.52760	0.00203	0.11482	0.00556	0.08697
1.0	0.52772	0.00212	0.11482	0.00552	0.08707
100	0.53662	0.00344	0.13311	0.00795	0.09994
250	0.54481	0.00480	0.14739	0.01054	0.10912
500	0.54724	0.00578	0.15388	0.01231	0.11407
750	**0.54624**	**0.00613**	**0.15550**	**0.01248**	**0.11460**

Если сравнивать лучший результат с диффузией без классификатора:

Метрика	Без классификатора	Guidance Scale = 750	Абсолютное изменение	Относительное изменение (%)
BERT-Score	0.52846	0.54624	+0.01778	+3.36%
BLEU	0.00252	0.00613	+0.00361	+143.25%
ROUGE-1	0.11700	0.15550	+0.03850	+32.91%
ROUGE-2	0.00611	0.01248	+0.00637	+104.26%
ROUGE-L	0.08938	0.11460	+0.02522	+28.22%