Exercise Briefing

Ki Hyun Kim

nlp.with.deep.learning@gmail.com

Objective:

- 모델 두 개를 동시에 학습할 수 있는 환경 갖추기
  - 학습 시작점 (e.g. dual_train.py, continue_dual_train.py)
  - Trainer (e.g. dual_trainer.py)
    - 입출력 tensor를 바꿔가며 반대쪽 모델에게 넣어줌
    - 예전 모델(e.g. Seq2seq, Transformer)을 그대로 활용 가능해야 함

- Loss 구현
  - 언어 모델 및 학습 환경 구현 (e.g. rnnlm.py, lm_trainer.py)
  - 각 모델의 backward() 호출을 위한 적절한 detach() 활용

Review: AutoGrad in PyTorch

- Computation Graph 동적 생성에 따른 AutoGrad
  - 필요에 따라 detach() 기능을 활용하여 back-prop을 통제할 수 있음

Equations

- LM을 미리 학습하여, 추후 freeze 한 상태로 활용

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$$

$$\hat{\phi} = \underset{\phi \in \Phi}{\operatorname{argmax}} \sum_{i=1}^{N} \log P(x_i; \phi)$$

$$\hat{\psi} = \underset{\psi \in \Psi}{\operatorname{argmax}} \sum_{i=1}^{N} \log P(y_i; \psi)$$

Equations

- detach()를 잘 활용하여, 각 loss를 적절히 계산해줘야 함

$$\theta_{x\to y} \leftarrow \theta_{x\to y} - \eta\nabla_{\theta_{x\to y}}\mathcal{L}(\theta_{x\to y})$$

$$\mathcal{L}(\theta_{x\to y}) = \sum_{i=1}^{N}\Bigg(-\log P(y_i|x_i;\theta_{x\to y}) + \lambda \times \Big(\big(\log P(x_i;\phi) + \log P(y_i|x_i;\theta_{x\to y})\big) - \big(\log P(y_i;\psi) + \log P(x_i|y_i;\theta_{y\to x})\big)\Big)^2\Bigg)$$

$$\theta_{y\to x} \leftarrow \theta_{y\to x} - \eta\nabla_{\theta_{y\to x}}\mathcal{L}(\theta_{y\to x})$$

$$\mathcal{L}(\theta_{y\to x}) = \sum_{i=1}^{N}\Bigg(-\log P(x_i|y_i;\theta_{y\to x}) + \lambda \times \Big(\big(\log P(x_i;\phi) + \log P(y_i|x_i;\theta_{x\to y})\big) - \big(\log P(y_i;\psi) + \log P(x_i|y_i;\theta_{y\to x})\big)\Big)^2\Bigg)$$

Equations

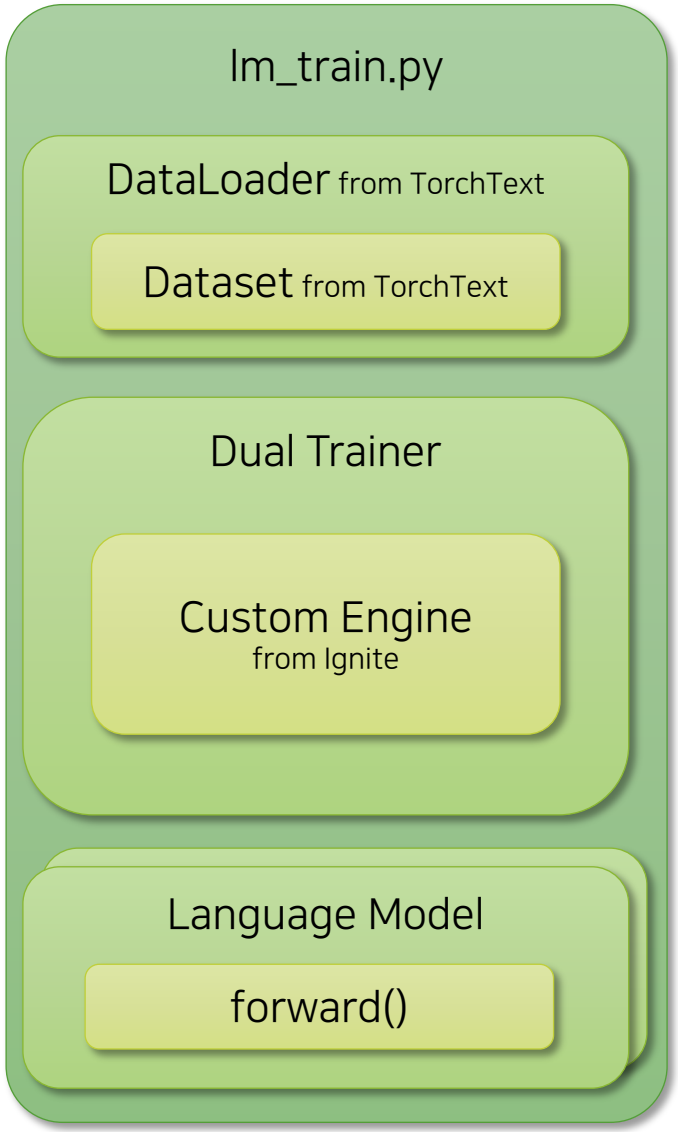
- 입출력 tensor쌍을 반대쪽 모델에 맞게 잘 변환해야 함

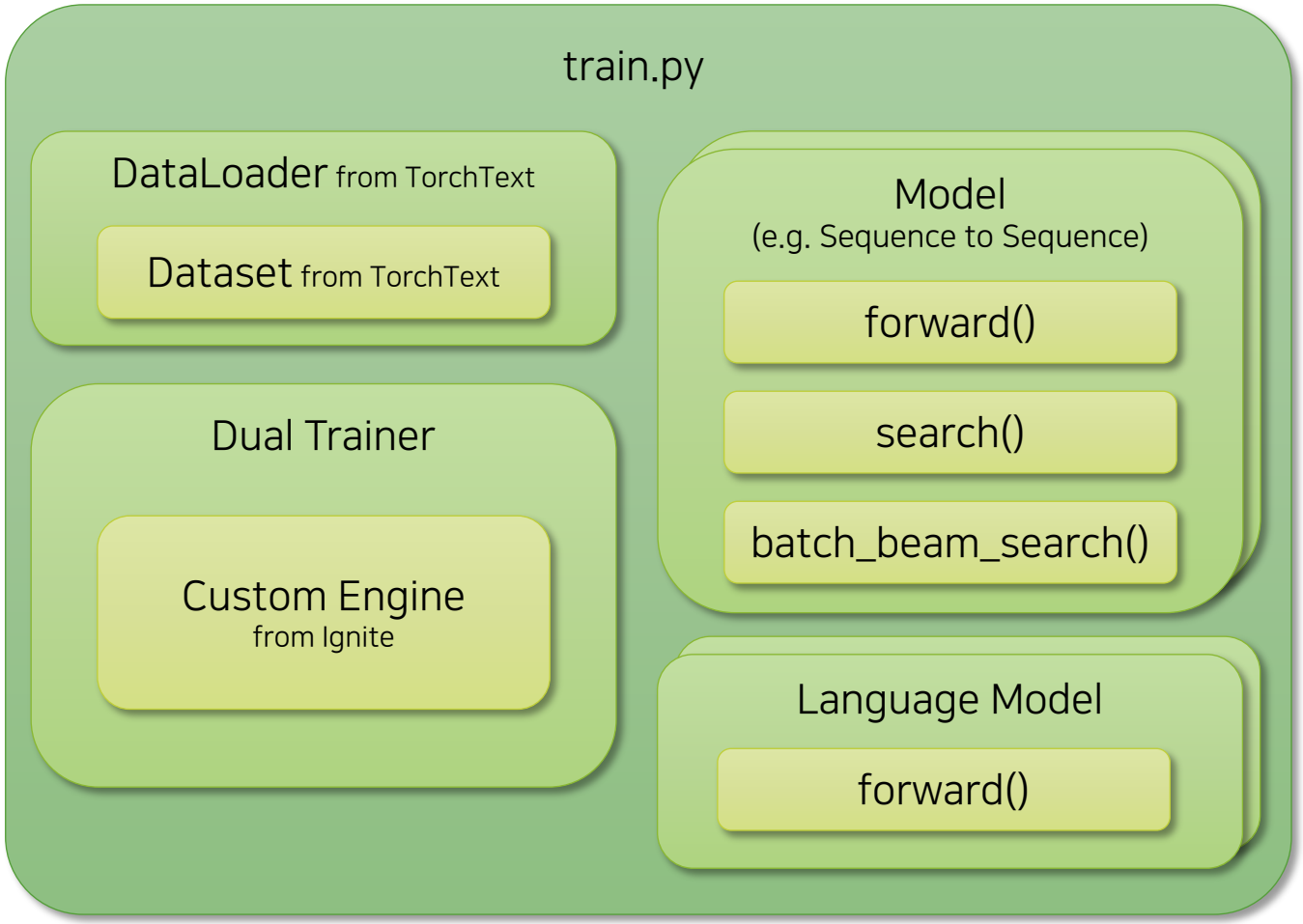
$$\log P(y_i|x_i;\theta_{x\to y}) = \sum_{t=1}^{m} \log P(y_{i,t}|x_i, y_{i,<t};\theta_{x\to y})$$
$$= \sum_{t=1}^{m} y_{i,t}^{\mathsf{T}} \cdot \log \hat{y}_{i,t}$$

$$\log P(x_i|y_i;\theta_{y\to x}) = \sum_{t=1}^{n} \log P(x_{i,t}|y_i, x_{i,<t};\theta_{y\to x})$$
$$= \sum_{t=1}^{n} x_{i,t}^{\mathsf{T}} \cdot \log \hat{x}_{i,t}$$

$$\text{where } x_{i,0} = y_{i,0} = <\text{BOS}> \text{ and } x_{i,n} = y_{i,m} = <\text{EOS}>.$$

Project Implementation





Training Procedure