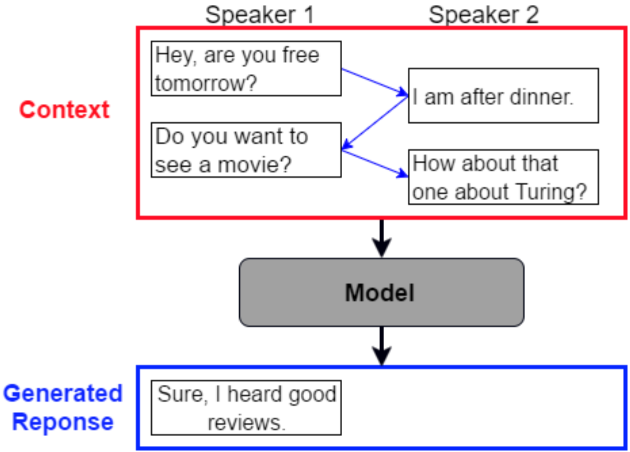
Policy Gradient Methods for Dialogue Response Generation

Nicolas A. Gontier   Michael Noseworthy

Reasoning and Learning Lab
McGill University

COMP 767 - Final Project Presentation
April 13th

Dialogue Generation



- We can measure the quality of a response using ADEM
  - A Dialogue Evaluation Model (R. Lowe, M. Noseworthy, I.V. Serban, N. Angelard-Gontier, Y. Bengio, and J. Pineau)

Dialogue Generation

- **Goal**: Train a model to maximize the ADEM score
- We will use the policy-gradient framework from RL
  - State (s_t): What has been generated up to this point $\hat{Y}_{1,\ldots,t-1}$ given a context c
  - Action (a_t): Emit a token [1] $\hat{w}_t$ in the generated response $\hat{Y}$ given a context c
  - Policy (π): The HRED [2] model (softmax over the vocab)
  - Return (R): The ADEM score for a generated response
    - Rewards are 0 except for the final step.
    - Reward part of sentences with ADEM might gives us a very bad signal
  - Work inspired by "An Actor-Critic Algorithm for Sequence Prediction" (D. Bahdanau et al., 2017)
- Data-set used: On-line Tweets (~700,000 conversations)

---

[1] We use BPE (sub-word level) tokens to reduce the size of the action space from ~20k to ~5k

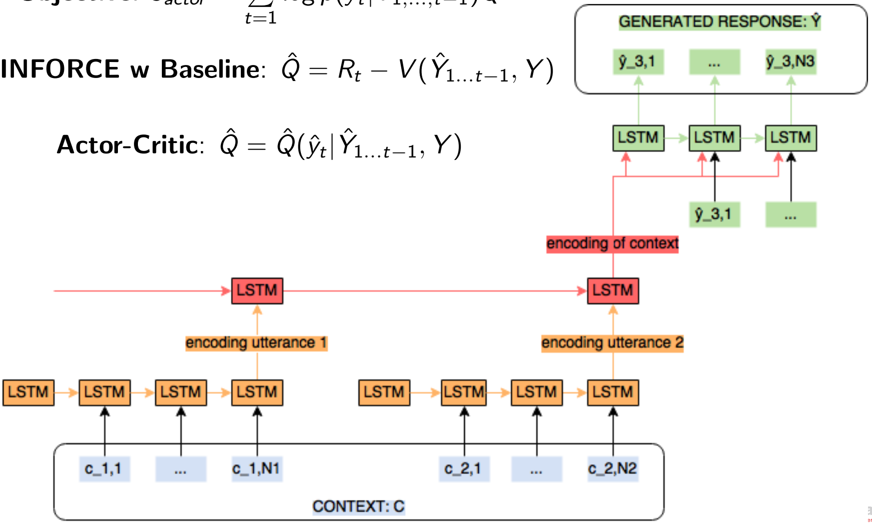
[2] I.V. Serban et al. (2016)

Actor Network

Objective: $J_{actor} = \sum_{t=1}^{T} \log p(\hat{y}_t | \hat{Y}_{1,\ldots,t-1}) \hat{Q}$

REINFORCE w Baseline: $\hat{Q} = R_t - V(\hat{Y}_{1\ldots t-1}, Y)$

Actor-Critic: $\hat{Q} = \hat{Q}(\hat{y}_t | \hat{Y}_{1\ldots t-1}, Y)$

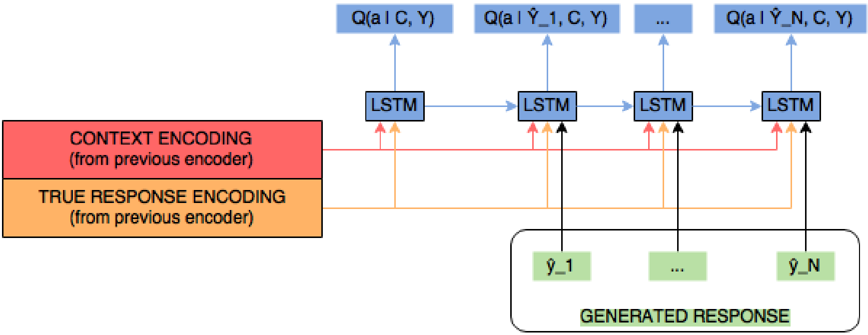


Critic Network

$$\textbf{TD Targets}: \; q_t = R(\hat{Y}_{1,\ldots,t}) - \bar{R} + \sum_{a \in Vocab} P(a|\hat{Y}_{1,\ldots,t}) * \hat{Q}(a|\hat{Y}_{1,\ldots,t})$$

$$\textbf{Objective}: \; J_{critic} = \sum_{t=1}^{T} (q_t - \hat{Q}(\hat{y}_t|\hat{Y}_{1,\ldots,t-1}))^2 + \lambda C_t$$

$$\textbf{Regularization}: C_t = \sum_{a \in Vocab} \hat{Q}(a|\hat{Y}_{1,\ldots,t-1}) - \bar{\hat{Q}}(.|\hat{Y}_{1,\ldots,t-1})$$



Challenges

Large Action Space

- Critic target q_t uses $(R_t - \bar{R})$ to reduce variance in the reward
- $J_{critic} = \sum_{t=1}^{T}$squared error loss$+\lambda C_t$ to penalize variance in the critic values $\hat{Q}(a|\hat{Y}_{1,\dots,t-1})$
- Pretrain the actor with ML objective: $J_{actor} = \sum_{t=1}^{T} \log p(\hat{y}_t | Y_{1,\dots,t-1})$
- Pretrain the critic with samples from the pretrained actor

Sparse Reward Signal

Things to try:

- Use ADEM to score sub-parts of generated response? May be really bad, takes more time.
- Monte Carlo roll-outs from each time steps to have a full sentence before sending it to ADEM? Very time consuming!